

DEVELOPMENT OF AN OPTIMIZATION PACKAGE : A COMPACT HEAT EXCHANGER APPLICATION

By
C. VIJAYA KUMAR

ME

984

TH
ME/1984/14
K96d

M

KUM

DEV



DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
MARCH, 1984

DEVELOPMENT OF AN OPTIMIZATION PACKAGE : A COMPACT HEAT EXCHANGER APPLICATION

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
C. VIJAYA KUMAR

to the
DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
MARCH, 1984

DEDICATED
TO
MY PARENTS
AS A MARK OF AFFECTION

10 JUL 1984*

ME-1984-M-KUM-DEV

83399

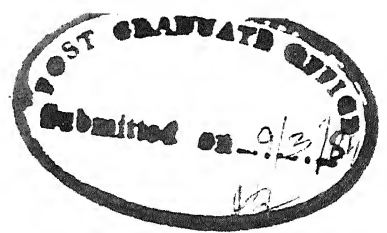
ACKNOWLEDGEMENTS

At the outset, I wish to express my deep sense of gratitude to Prof. H.C. Agrawal for his valuable guidance. I am highly grateful to him for his understanding, keen encouragement, kind help extended to me in carrying out this project.

It is impossible to fully appreciate in words the warmth and affection that my innumerable friends accorded to me throughout the period of my stay here.

I personally thank all the people who have associated with me in all dimensions and in all spaces.

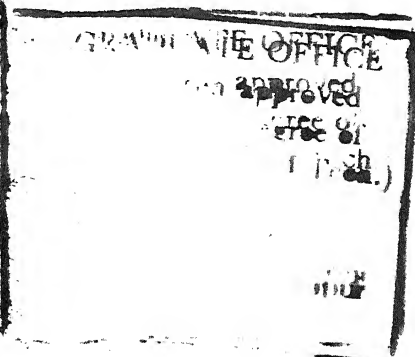
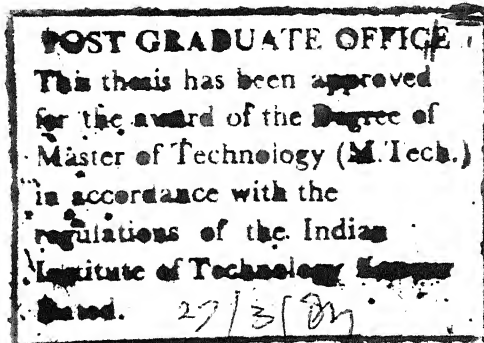
- C. VIJAYA KUMAR



CERTIFICATE

Certified that the work entitled "DEVELOPMENT OF AN OPTIMIZATION PACKAGE : A COMPACT HEAT EXCHANGER APPLICATION" by C. VIJAYA KUMAR has been carried out under my supervision and has not been submitted elsewhere for a degree.

DR. H.C. AGRAWAL
PROFESSOR
DEPT. OF MECHANICAL ENGG.
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR



ABSTRACT

A package of general and efficient computer code for solving optimization problems is presented in this thesis. Powerful unconstrained optimization methods based on the ideas of quadratic convergence and conjugate directions are used. Constrained optimization problems are solved, using the penalty functions and feasible directions method.

These programs are tested for convergence and efficiency by using typical test functions previously suggested in the open literature. Results of a comparative study of constrained optimization methods indicate that sequential penalty function methods, with a gradient-based search procedure, using finite difference gradient computations, perform well on most problems.

The capability of the package is well demonstrated by the design of a direct transfer air-to-water compact heat exchanger employing flat-finned tube surface, the objective being minimum pumping power and the constraints imposed on volume, dimensions of the exchanger and heat transfer. The ϵ -NTU method is used for evaluating the heat exchanger performance. Polynomial expressions

are found to approximate the experimental data for the fluid transport properties and the friction and colburn factors for the heat transfer surface. This optimization study of the compact heat exchanger provides a design which has got much superior performance.

LIST OF FIGURES

Fig. No.		<u>Page</u>
1	Optimization Techniques	14
2a	Flow Diagram for Unconstrained Optimization	15
2b	Flow Diagram for Constrained Optimization	16
3	Heat Transfer Surface Area Density Spectrum of Exchanger Surfaces	30
4	Heat Exchanger Optimization Methodology	36
5	Heat Exchanger Configuration	43
6	Friction and Colburn Factors for Air side	45
7	Friction and Colburn Factors for Tube side	46
8	Performance Analysis Chart	50
9	Expressions for Fluid Properties, f and j	56
10	Variation of Effectiveness with Volume	62
11	Variation of Pumping Power with Volume	63
12	Trade-off Curve	67
13	Variation of Effectiveness with Allowable Volume	68
14	Optimum Flows	69

TABLE OF CONTENTS

	<u>Page</u>
CERTIFICATE	
ACKNOWLEDGEMENTS	
ABSTRACT	
CHAPTER-1 : INTRODUCTION	1
1.1 : HEAT EXCHANGER OPTIMIZATION	2
1.2 : OPTIMIZATION	4
1.3 : PREVIOUS WORK	7
1.4 : PRESENT WORK	10
CHAPTER-2 : PACKAGE STRUCTURE	13
2.1 : INTRODUCTION	13
2.2 : RANGE FINDER	17
2.3 : ONE DIMENSIONAL SEARCH METHODS	19
2.4 : MULTI DIMENSIONAL UNCONSTRAINED SEARCH METHODS	20
2.5 : MULTI DIMENSIONAL CONSTRAINED SEARCH METHODS	21
2.5.1 Interior Penalty Function Method	22
2.5.2 Exterior Penalty Function Method	23
2.5.3 Feasible Directions Method	24
2.5.4 Convergence Rule	25

CHAPTER-3	:	COMPACT HEAT EXCHANGER	27
3.1	:	INTRODUCTION	27
3.2	:	CONSTRUCTION TYPES AND SURFACE GEOMETRIES	32
3.3	:	HEAT EXCHANGER DESIGN METHODOLOGY	33
CHAPTER-4	:	OPTIMUM DESIGN OF A COMPACT HEAT EXCHANGER	38
4.1	:	INTRODUCTION	38
4.2	:	NOMENCLATURE	39
4.3	:	PROBLEM FORMULATION	42
		4.3.1 Design Problem	42
		4.3.2 Selection Aspects	42
		4.3.3 Optimization Problem Definition	44
4.4	:	PERFORMANCE CALCULATION PROCEDURE	49
		4.4.1 Overview	49
		4.4.2 Core Dimensions	51
		4.4.3 Outlet Fluid Tempera- tures	52
		4.4.4 Fluid Properties	53
		4.4.5 Reynolds and Prandtl Number	54
		4.4.6 Friction and Colburn Factors	55
		4.4.7 Heat Transfer Film Coefficients	55
		4.4.8 Fin Effectiveness	57
		4.4.9 Exchanger Effectiveness	57
		4.4.10 Pressure Drops	58

4.5	:	SOLUTION TECHNIQUES	60
CHAPTER-5	:	RESULTS AND DISCUSSION	65
CHAPTER-5	:	TRADE-OFF STUDIES	65
5.2	:	A COMPARISON OF OPTIMIZATION METHODS	70
5.3	:	CONCLUSIONS	72
REFERENCES			73
APPENDIX-A	:	USERS MANUAL	
APPENDIX-B	:	PROGRAMS PACKAGE LISTING	

CHAPTER 1

INTRODUCTION

In modern technology, it is becoming increasingly important for engineering designs of all types to offer the best possible performance. To achieve this task, the analyst must identify the variables which influence the behaviour of the system, judiciously select quantitative measures that appropriately describe the system performance and then make a detailed parametric study. Frequently, it is found that some of the desired objectives have conflicting trends. Also, in many applications it is not possible to keep the design variables constant as they may have a tendency to drift with time or fluctuate between some limits. For a reliable engineering design it is not adequate to just determine the best operating conditions. It is equally important to know how sensitive the optimum performance is, to changes in the system parameters. Another important consideration that must be kept in mind while making a design study is that the design requirements may change in future, and it would be very useful if a suitable alternate design can be predicted from the results of the current optimum. This

would eliminate the need for making another extensive parametric study. This necessitates the introduction of a design tool that will permit the engineer to experiment with design variables and different performance criteria, to search for the best operating conditions and to identify the influence of the system parameters as the selected performance indices. It is also desirable to include some aids to show how the design specifications constrain the system performance.

Computer-aided design procedures are playing an important role in the design practice, making it possible to analyze a wide variety of complex problems. Recent advances in computer-graphics, real time manipulation capabilities along with the developments in optimization theory and interactive computation have made it possible to tap the design tools' potential. The present work is concerned with the development of a single package which includes several optimization methods. The package has been used for optimum design of a compact heat exchanger.

1.1 HEAT EXCHANGER OPTIMIZATION

Heat exchangers are designed for many different applications and hence may involve many different performance criteria. These criteria may be:

minimum initial cost, minimum initial and

operating costs, minimum weight or material, minimum volumes or heat transfer surface area, and so on.

When a single performance measure has been defined quantitatively and is to be minimized or maximized, it is called an "objective function" in a design optimization. A particular design may be subjected to certain "CONSTRAINTS" such as desired heat transfer rate, allowable pressure drop, limitations on height, width and/or length of the exchanger. A number of different surfaces could be incorporated in a specific design problem and there are many geometrical parameters that could be varied for each surface geometry. For an extended surface exchanger, the geometrical variables associated with a fin are the fin pitch, fin height, fin thickness, type of fin, and other variables associated with each fin-type. In addition, operating flows and temperatures could also be changed. Thus a large number of "design variables" are associated with a heat exchanger design. The question arises as to how one can effectively adjust these design variables within imposed constraints and come up with a design having optimum objective function. This is what we mean by the most "efficient" design.

A complete mathematical optimization of heat exchanger design is neither practical nor possible.

Many engineering judgements based on experience are involved in different stages of the design. However, once the general configuration and surfaces are selected for a specific application, an optimized heat exchanger design may be arrived at if the objective function and constraints can be expressed mathematically, and if all of the variables are automatically and systematically changed on some statistical or mathematical basis.

A large number of optimization (search) techniques are available in literature. The difficulty with the present state of optimization art is that while any single search technique may work well on many problems, no single technique is able to solve every problem. A package of optimization programs is a convenient approach to overcoming this difficulty. Well developed, the programs can be used interchangeably and a new search can be called in, where a previous one has quit thereby continuing or assuring progress toward optimum. In addition, when a package is used, various parts of the search procedure common to several optimization routines need be programmed only a single time.

1.2 OPTIMIZATION

Optimization techniques offer systematic approach for seeking the best operating conditions. A description

of many of the more popular optimization algorithms is presented in [1, 2, 3]. Almost all design problems require either the maximization or the minimization of some parameter. This parameter is usually called the design objective function. For example the problem may call for a heat exchanger with a minimum volume. The expression for volume would be the design objective function. This is done by selecting values for certain variables. These variables are known as design or decision variables. For the design to be acceptable, it must satisfy certain constraints. For example, an air heater must be designed so that it will fit into a given space.

The general nonlinear constrained optimization problem can be written mathematically as:

$$\text{Minimize} \quad F(\bar{X}) \quad (1)$$

$$\text{Subject to:} \quad g_i(\bar{X}) \leq 0 \quad i = 1, 2, \dots, m \quad (2)$$

$$l_k(\bar{X}) = 0 \quad k = 1, 2, \dots, p \quad (3)$$

$$x_q^l \leq \bar{x}_q \leq x_q^u \quad q = 1, 2, \dots, n \quad (4)$$

Where the vector, \bar{X} , is the vector of "n" design variables, F is the objective function, g_i is a set of "m" inequality constraints, l_k is a set of "p" equality constraints and the lower and upper bounds x_q^l , x_q^u are the side constraints imposed on the design variables. Side constraints could also be included in (2), but for

better efficiency they are treated separately by some software experts. Equations (2) and (3) may be linear or nonlinear functions of the design variables. They may be explicit or implicit functions of \bar{X} , but must have continuous first derivatives. The basic solution procedure is to, systematically, search the design space by sequentially varying the design parameters along search directions that reduce the objective function. For example, starting at a point \bar{X}_q defined by the existing values for the design variables, a new point \bar{X}_{q+1} is determined as:

$$\bar{X}_{q+1} = \bar{X}_q + \alpha \bar{S}_q$$

$$F(\bar{X}_{q+1}) < F(\bar{X}_q)$$

Where \bar{S}_q is the search direction, and α is the step size along that direction. The essence of nonlinear programming methods lies in determining the α which minimizes $F(\bar{X}_{q+1})$ by using a one-dimensional search technique. A new search direction is computed at \bar{X}_{q+1} . A minimum is obtained if it satisfies the accuracy criterion. There is no evidence that the resulting minimum is the global optimum, since relative minima may be present. The search procedure must be repeated using different starting points if this is a possibility. For problems based on real processes, luckily, the objective function is usually a

well behaved function with a single extremum. Therefore, for most practical purposes, the use of standard numerical procedures that provide a local solution to the optimization problem is not a great disadvantage.

1.3 PREVIOUS WORK

Compact heat exchanger design is a complex task requiring the examination and optimization of a wide variety of heat transfer surfaces. Smith [4] has listed some typical advantages of direct cooling with air as compared to cooling with water in a shell-and-tube exchanger. Studies have shown that a poor choice of either the heat transfer surfaces or design parameters can more than double the costs chargeable to a heat exchanger.

The method of Bergles et al. [5] is primarily meant for comparing the performance of heat exchanger surfaces for some specified criteria by adjusting only two exchanger variables. Their method does not include any minimization technique but results show that a great improvement in heat exchanger performance can be made by proper selection of design parameters.

Fax and Mills [6] developed a method, using Lagrange multipliers to optimize heat exchanger design under specified constraints. This technique requires

the objective function and the constraints expressed in explicit equation form and differentiable throughout the range of interest, the total number of constraints be less than the total number of variables, and all constraints be equality constraints. Obviously, the use of this method is restricted to a limited number of problems. Briggs and Evans [7] recommended a "logic search method" in which an experienced engineer makes use of selecting design variables based on his apriori knowledge to obtain an optimum design. This method is less scientific and can be tried only if the computer time and storage capacity are freely ensured. Wilde [8] has surveyed various optimization techniques applicable to heat exchanger design. He focussed on the geometric programming technique and other search methods which require explicit expressions for the constraints and the objective function. The geometric programming technique also requires that all of the constraints and the objective function must be expressed in a particular power law equation form. These methods have limited applications for a general heat exchanger optimization problem.

Palen et al. [9] proposed the Complex Method [1], for the heat exchanger optimization problem. They found a minimum shell-and-tube exchanger by varying six geometrical parameters. The Complex Method requires

several feasible starting designs before optimization can be performed. Johnson et al. [10] coupled an existing shell-and-tube condensor design code with a constrained function minimization code to produce an automated marine condensor design program of vastly different complexity.

Fontein and Wassink [11] utilized the Simplex Method [1] and steepest descent method for optimizing a shell-and-tube exchanger. Shah et al. [12, 13] have initiated work on package approach lines to the optimization problems in compact heat exchangers. The package includes the numerical nonlinear programming techniques. Hedderich et al. [14] have also developed a massive code using the method of feasible directions and the augmented Lagrange multiplier method. It has been used for the design of an air-to-water finned-tube heat exchanger.

It is experienced that, although, there are many methods that have been presented for heat-exchanger optimization, each of the methods has its own limitations; none is completely general. Among the design procedures cited above, those which are applicable, to cross flow, air cooled compact heat exchangers are limited to the 120 individual surfaces found in the open literature [15] for the calculation of the air side heat-transfer coefficient and friction factor. Therefore, the designer is faced with choosing an optimum surface from a number

of individual optimal designs calculated from one of the above methods.

1.4 PRESENT WORK

The work embodied in this thesis is an attempt in two directions:

- (i) to develop a general purpose computer aided design procedure that assists the engineer in establishing a best design. This makes available to the user, numerous nonlinear programming techniques. The search parameters can be changed and the user can switch from one search technique to another. This flexibility, provided to the user, gives a feel for the various techniques applicability and to select the most suitable one to the problem in hand. The development of this computer package is described in the next chapter.
- (ii) to use the above package for optimizing the design of a compact heat exchanger for some specific objective and constraints. A direct transfer, air-water, finned-tube heat exchanger for a gas turbine plant has been optimized. The objective is to minimize pumping power requirements of the heat exchanger which has a volume

less than a specified value and for which the heat transfer rate is prescribed. The flow rates of the fluids and the linear dimensions of the exchanger comprise the design variables of the problem. Fixed bounds on these variables are also imposed as side constraints. Constraints take into account the aesthetic value to get a realistic configuration for the exchanger.

The package implements the following multi-dimensional minimization schemes: univariate, steepest descent, conjugate directions, conjugate gradient, variable metric method [1,2,3]. These methods use either a quadratic interpolation procedure or golden section technique [1,2,3] for 1-D minimization. Constrained problems are solved using either penalty function or feasible directions [1,3] method. To check the reliability and flexibility of the proposed package, a number of problems available in literature have been solved and results compared and listed in Appendix-A. These problems differ fundamentally in the manner in which the objective functions and the constraints are computed in the function subroutine. These quantities are usually programmed in the user written function subroutine as follows:

1. Closed form algebraic expressions
2. Mostly algebra but requiring some iterations

The programs are simple to use, with the user having to provide a function subroutine and to specify a minimum number of variables and flags. The ensuing chapter deals with the package structure that has been developed.

CHAPTER 2

PACKAGE STRUCTURE

2.1 INTRODUCTION

A set of general programs have been developed which make available to the user, reliable mathematical programming routines suitable for optimizing a large class of problems. Various numerical techniques which have been utilized to structure this optimization package are briefly discussed in this chapter and summarized in Fig. 1. Flow diagrams for solving unconstrained and constrained optimization problems are given in Fig. 2. Complete details regarding the algorithms used, may be obtained from [1,2,3]. Appendix-A gives the details for using the programs with examples to illustrate the procedure and program-listings have been given in Appendix-B.

An interesting feature of the package is it's flexibility. New search methods can be easily introduced by making minor changes. This is made possible by having maximum interaction between the subroutines and dividing the procedure into the following four basic operations:

1. Selection of a search direction
2. Estimation of the range for minimum along the search direction

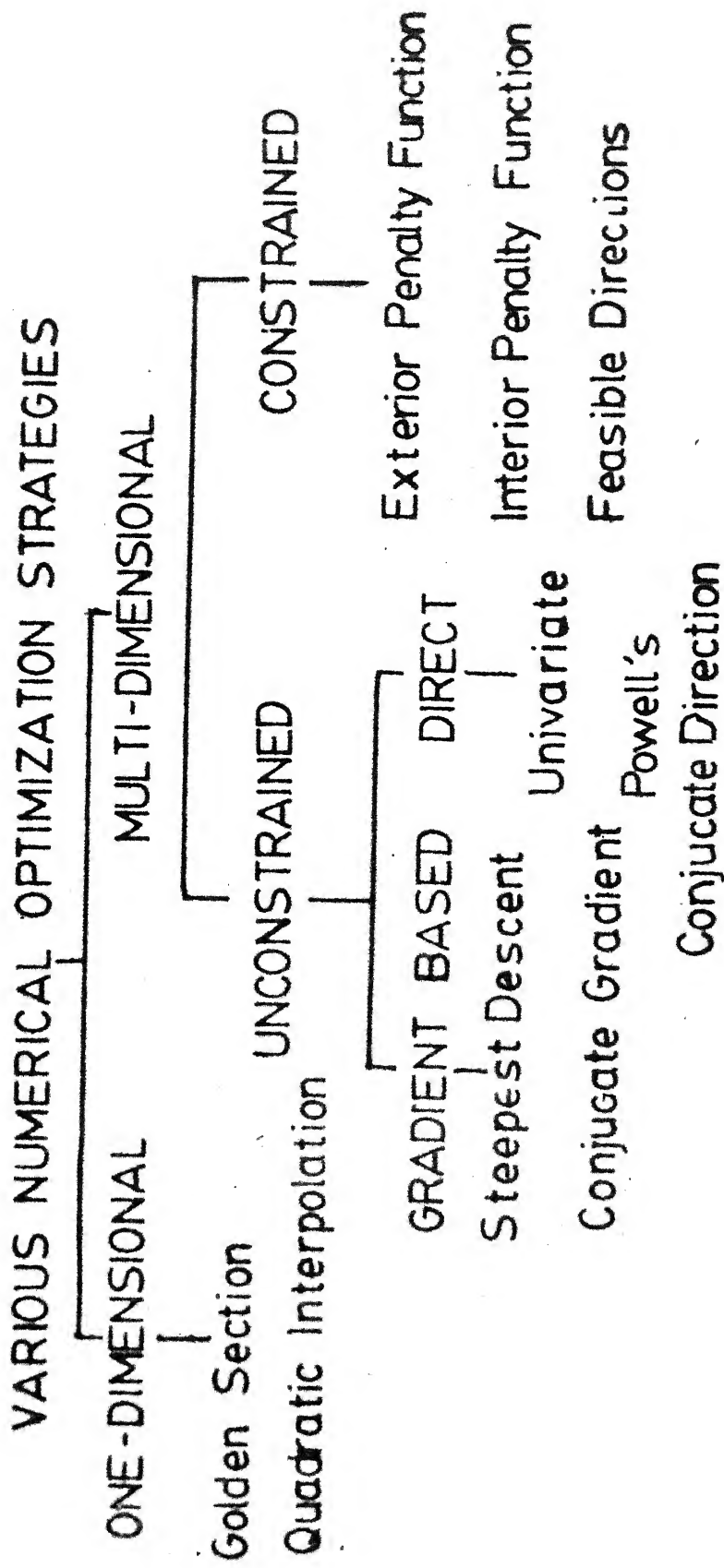


Fig1. Optimization Techniques

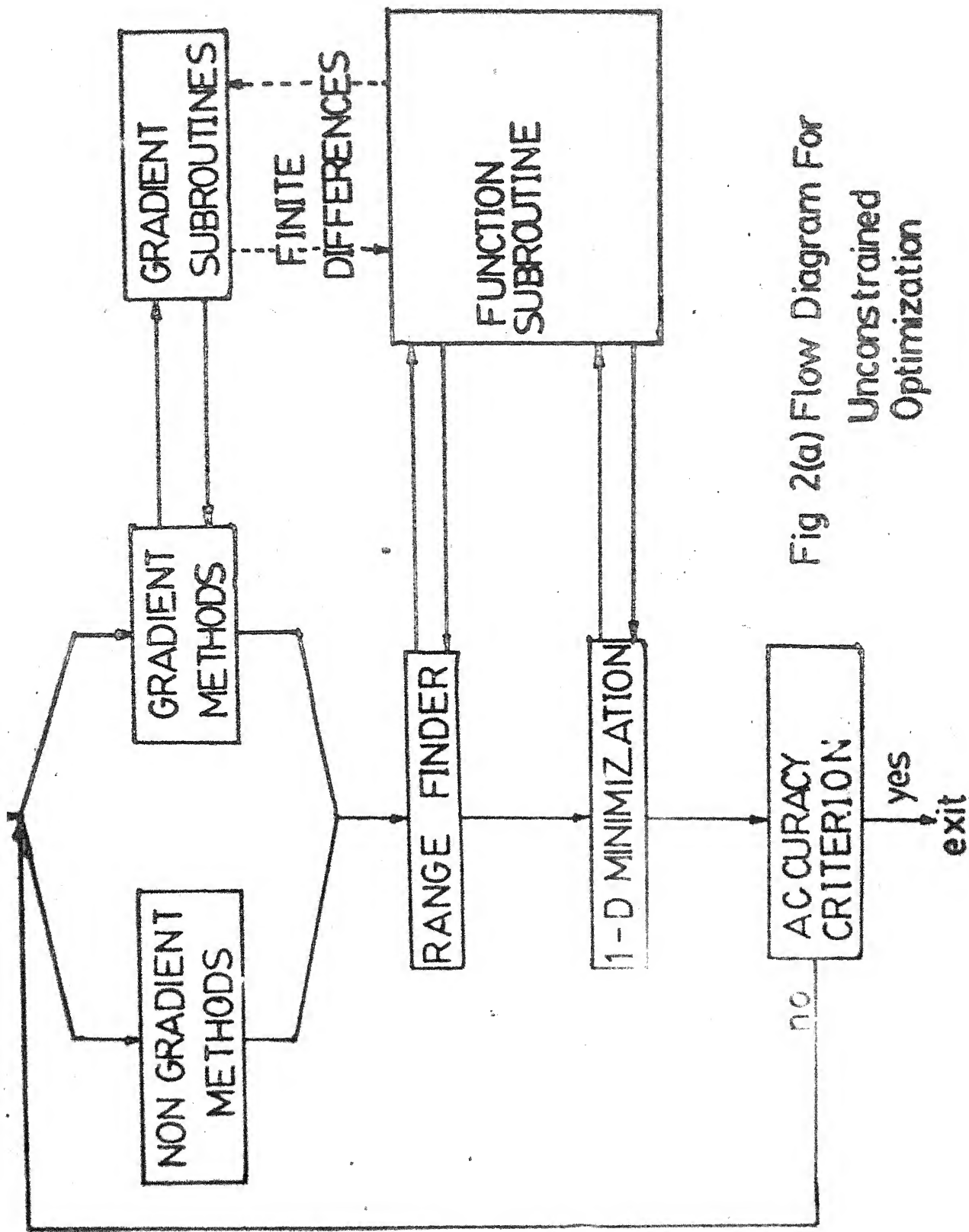


Fig 2(a) Flow Diagram For
Unconstrained
Optimization

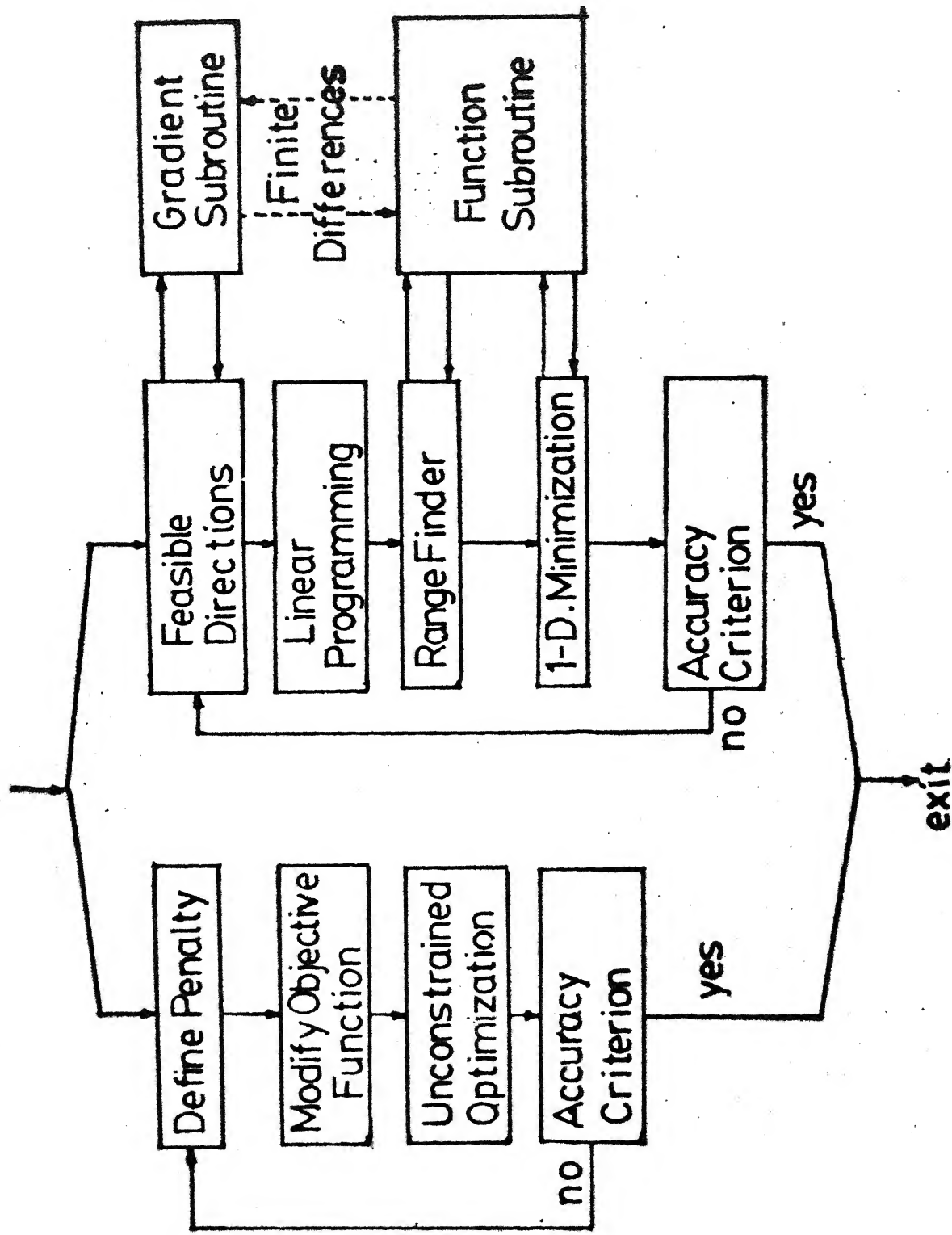


Fig 2(b). Flow chart for Constrained Optimization

3. One-dimensional minimization to locate the minimum along the search direction
4. Checking for convergence

The multidimensional methods differ primarily in the manner by which the search direction is determined. The effectiveness of the methods for computing a given search direction depends on the accuracy with which the minimum along previous search directions has been obtained. Thus, the one-dimensional minimization influences the stability and convergence of the multi-dimensional methods.

2.2 RANGE-FINDER: SUBROUTINE RANGE

Nonlinear programming methods essentially consist of a series of unidimensional minimizations. Numerical solution procedures for locating a minimum in a search direction require a prior knowledge of the limits between which the minimum exists. These limit points are determined by the range-finder by sequentially increasing the step-size along the search direction until three consecutive points are obtained, such that

$$\begin{aligned} F(\bar{X}_q + \alpha_1 \bar{S}_q) &> F(\bar{X}_q + \alpha_2 \bar{S}_q) \\ F(\bar{X}_q + \alpha_3 \bar{S}_q) &> F(\bar{X}_q + \alpha_2 \bar{S}_q) \end{aligned} \tag{5}$$

where α_1 , α_2 , α_3 define three consecutive points along the search direction and \bar{X}_q is the design vector the \bar{S}_q is search direction. The Equation (5) guarantees

that a minimum exists between α_1 and α_3 .

The inputs to RANGE are \bar{X}_q , \bar{S}_q , initial step-size α , maximum increment in stepsize and the number of iterations, permitted to determine α_1 , α_2 , and α_3 . It is desirable to obtain a small value for $(\alpha_3 - \alpha_1)$ in a few iterations. If a step along \bar{S}_q reduces F , α is doubled for the next iteration, and the values for α_1 and α_2 updated. The increment in α is not allowed to exceed the maximum specified value. When more iterations are required to estimate $(\alpha_3 - \alpha_1)$ than that specified, RANGE assumes the last value of α to be the location of the minimum along \bar{S}_q . While seeking α_2 , RANGE has the provision to reverse the search direction and reduce α , if an initial step along \bar{S}_q increases F .

Since RANGE is used also by constrained optimization methods, some additional features have to be introduced to take care of constraints. When using the method of feasible directions, it is necessary to locate constraints that are violated. If the point \bar{X}_{p+1} violates constraint "g", RANGE repetitively uses a linear interpolation scheme to find α' , where

$$\alpha' = \frac{\alpha_p g(\bar{X}_{p+1}) - \alpha_{p+1} g(\bar{X}_p)}{g(\bar{X}_{p+1}) - g(\bar{X}_p)} \quad (6)$$

$$\bar{X}_p = \bar{X}_q + \alpha_p \bar{S}_q$$

$$\bar{X}_{p+1} = \bar{X}_q + \alpha_{p+1} \bar{S}_q$$

where $g(\bar{x}_p)$, $g(\bar{x}_{p+1})$ are constraint function such that $g(\bar{x}_q + \alpha' \bar{s}_q) \leq \epsilon$ - the accuracy needed.

In using the interior penalty function method, the search is restricted to the feasible region. When \bar{x}_{p+1} violates "g", RANGE progressively reduces the step size to find α' , which is now expressed as:

$$\alpha' = \alpha_p + \frac{\alpha_{p+1} - \alpha_p}{2} \quad (7)$$

such that $g(\bar{x}_q + \alpha' \bar{s}_q) \leq 0$

2.3 ONE-DIMENSIONAL SEARCH METHODS

The range-finder determines the interval which contains the minimum along a search direction. An efficient unidimensional search is next, needed to accurately locate the minimum in a few iterations. The 1-D search problems are of the form:

$$\text{minimize } \phi(x)$$

where x is a single variable whose optimum value is to be determined, so that the function ϕ is at its minimum.

The program package gives the user, the option of using either the quadratic interpolation scheme (SUBROUTINE QUAD) or the golden section method (SUBROUTINE GOLD) [13]. The former performs a series of iterations approximating $\phi(x)$ as a quadratic function; the latter isolates the minimum in regions of successively decreasing

size. The number of iterations needed for convergence to the minimum, will depend on the accuracy needed. Compared to quadratic interpolation scheme, the golden section method converges slowly to the minimum.

2.4 MULTIDIMENSIONAL UNCONSTRAINED SEARCH METHODS

Unconstrained searches, i.e., to just minimize $F(\bar{X})$ without any constraints, can be performed by a sequence of one-dimensional minimizations in appropriate directions. The most efficient search procedures generate directions that are conjugate.

In direct search strategies, the search directions are established apriori but may be modified during the course of a search. In the univariate method (SUBROUTINE UNIV), the design space is searched along the coordinate directions, every time adjusting one variable and keeping others constant. A cycle is completed when all the coordinate directions are searched and repeated, if needed, to satisfy the accuracy criterion. Powell's method (SUBROUTINE CONDIR) begins with the univariate approach but establishes "conjugate" directions by moving along vectors connecting points in the design space, resulting from previous 1-D minimizations. Conjugate directions [1,2,3] lead to speeded convergence of an optimization on most functions and, infact, guarantee convergence on

an n -dimensional quadratic function after n , 1-D minimizations in mutually conjugate directions.

Gradient based search techniques utilize gradient information in establishing directions for 1-D minimizations. The objective function has a maximum rate of change along this direction. Thus, moving in the negative gradient direction would seem to reduce the performance index most (SUBROUTINE STEEP). The conjugate gradient (SUBROUTINE CONGRA) and the Davidon-Fletcher-Powell (SUBROUTINE DFPM) methods [1,2,3] utilize the past and current gradient values to generate conjugate directions and speed the convergence. The Davidon-Fletcher-Powell approach is generally considered to be one of the most effective methods for unconstrained optimization. Difficulties seldom arise except on very badly distorted or eccentric functions. When the search direction fails to reduce the function, the search procedure is restarted after adjusting certain parameters accordingly.

2.5 MULTI-DIMENSIONAL CONSTRAINED SEARCH METHODS

The constraints in an optimization problem can be enforced either by using substitutions [1,3] to eliminate them or by modifying the objective function [1,2,3] with penalty terms. The penalty terms increase the objective function when a constraint is violated.

Substitutions and penalty terms transform the constrained problem into a sequence of unconstrained problems. Alternate procedures for obtaining the constrained minimum are based on finding search directions [1,3] that are directed towards the feasible region and away from the constraints just violated. The search for the optimum is confined to the feasible region.

2.5.1 Interior Penalty Function Method

SUBROUTINE IPENAL

The basic idea underlying this method is to solve repetitively a sequence of unconstrained problems whose solutions, in the limit, approach the minimum of the constrained optimization problem. The objective function is modified as

$$\phi(\bar{X}, R) = F(\bar{X}) - R \sum_{j=1}^m \frac{1}{g_j(\bar{X})} + \frac{1}{\sqrt{R}} \sum_{k=1}^l [l_k(\bar{X})]^2$$

..... (8)

where ϕ is the modified function called PENALTY, R is penalty factor, "m" is the number of inequality constraints, and "l" is the number of equality constraints. The penalty term with inequality constraints is small for points away from the constraints but blows up as the constraints are approached. As R is decreased, the penalty term, with equality constraints l_k , blows up and it forces l_k to be satisfied to some degree. When

g_j and l_k are satisfied:

$$\phi(\bar{X}^*, R) \rightarrow F(\bar{X}^*) \text{ as } R \rightarrow 0 ,$$

\bar{X}^* is the optimum solution.

An extrapolation scheme suggested by Fiacco and McCormick [1,2,3] is used to predict a new starting point and the value of the constrained optimum. The whole search should be strictly within the feasible region, because a mirror image exists in the infeasible region. Thus, precaution is taken in RANGE so as not to overstep into the infeasible region. An appealing feature of this method is that it produces an improving sequence of acceptable designs; hence allowing the designer to compare sub-optimal designs to the optimum one.

2.5.2 Exterior Penalty Function Method

SUBROUTINE EXPEN

A frequently used exterior penalty function which accommodates inequality and equality constraints is of the form

$$\phi(\bar{X}, R) = F(\bar{X}) + R \sum_{j=1}^m \langle g_j(\bar{X}) \rangle^2 + R \sum_{k=1}^1 [l_k(\bar{X})]^2$$

..... (9)

where $\langle g_j(\bar{x}) \rangle = 0$ for $g_j(\bar{x}) \leq 0$
 $= g_j(\bar{x})$ for $g_j(\bar{x}) > 0$

$\phi(\bar{X}, R)$ is minimized for increasing values of R . As R

is increased, the terms involving g_j and l_k become bigger. The \bar{X} vector is forced by the penalty terms to satisfy the constraints to some degree. As long as g_j and l_k are satisfied, as $\bar{X} \rightarrow \bar{X}^*$, the value of the penalty becomes negligible:

$$\phi(\bar{X}^*, R) \rightarrow F(\bar{X}^*)$$

X^* is the constrained optimum solution. This method converges to the constrained optimum from outside the feasible region for a sequence of unconstrained minimizations with increasing R .

Selection of proper initial values of R depends on the design problem and one's judgement. Fox [1], Rao [3] discussed some rules for picking an initial value of R . A source of trouble in the penalty function methods lies in the relative magnitudes of the constraints. If $g_1 = 1000 g_2$, g_1 changes more rapidly than g_2 , hence, overpowers it over the infeasible region. ϕ is insensitive to g_2 , and the optimum may even violate g_2 . This problem is overcome by scaling the constraints.

2.5.3 Feasible Directions Method

SUBROUTINE FEAS

This is based on Zoutendijk's procedure and begins at a point within the constraints, using an unconstrained optimization procedure until a constraint

is encountered. Then, a usable feasible direction \bar{S} is defined, which reduces F and does not violate the constraints, by solving the following linear programming problem:

$$\begin{array}{ll} \text{Max } \beta & \text{subject to} \\ \bar{S}, \beta & \left. \begin{array}{l} \bar{S}^T \nabla F + \beta \leq 0 \\ \bar{S}^T \nabla g_j + \theta_j \beta \leq 0 \quad j \in J \end{array} \right\} (10) \\ & |s_i| \leq 1 \quad i = 1, 2, \dots, n \end{array}$$

where β is a slack variable, T indicates transpose, J is the set of active constraints and the s_i are the components of \bar{S} . The θ_j are the "push-off" factors and, as suggested by Fox, are set to unity for nonlinear constraints and zero for linear constraints.

2.5.4 Convergence Rule

SUBROUTINE CONVRG

For the multi-dimensional search techniques an optional convergence rule is provided. Besides satisfying the accuracy criterion, this option perturbs the design variables and uses a scheme similar to Powell's [1]. It works as follows:

1. Apply the search procedure till the accuracy is achieved. Call the resultant point \bar{A} .
2. Perturb the design variables by 1%

3. Apply the normal search until the accuracy is achieved. Call the resultant point \bar{B} .
4. Terminate if $F(\bar{A})$ and $F(\bar{B})$ differ by the accuracy desired, or else search for minimum along the line joining \bar{A} and \bar{B} and starting from a point having lower performance index (say \bar{A}); call it \bar{C} and neglect \bar{B} .
5. Terminate if $F(\bar{C})$ and $F(\bar{A})$ satisfy the accuracy criterion, or else repeat step 1 using \bar{C} as starting point.

The above scheme is less likely to stop prematurely, but it may prove to be time consuming.

Chapters 3 and 4 describe a compact heat exchanger and its design methodology to help understanding how the optimization package, developed in this chapter, may be used for optimizing the design of a compact heat exchanger.

CHAPTER 3

COMPACT HEAT EXCHANGER

3.1 INTRODUCTION

A heat exchanger is a device which provides for transfer of ~~internal~~ thermal energy between two or more fluids at differing temperatures. Heat transfer between fluids takes place through a separating wall. Since the fluids are separated by a heat transfer surface, they do not mix. Common examples of such heat exchangers are the shell-tube-exchangers, automobile radiators, condensers, evaporators, air preheaters. A heat exchanger consists of the active heating elements such as a core or a matrix containing the heat transfer surface, and passive fluid distribution elements such as headers, manifolds, tanks, inlet and outlet nozzles or pipes, or seals. Usually, there are no moving parts in a heat exchanger; however, there are exceptions such as a rotary regenerative exchanger, in which the matrix is mechanically driven to rotate at some design speed.

The heat transfer surface is the surface of the exchanger core which is in direct contact with the fluids

and through which heat is transferred by conduction. That portion of the surface which also separates the fluids is referred to as "primary or direct surface". The design thermal effectiveness of exchangers employing such surfaces is usually 60% and below and the heat transfer surface area density is usually less than $300 \text{ m}^2/\text{m}^3$. In many applications, a much higher, upto 98% , exchanger effectiveness is essential, and the box volume and mass are limited so that a much more compact surface is mandated. Usually, either a gas or a liquid having a low heat transfer coefficient is the fluid on one or both sides. This results in a large heat transfer surface area requirements. For low density fluids (gases), pressure drop constraints, tend to require a large flow area. So a question arises, how can we increase both the surface area and flow area together in a reasonably shaped configuration. Appendages or fins on the primary surfaces increase the surface area density. Flow area is increased by the use of thin gauge material and sizing the core properly. The heat transfer coefficient on the extended surfaces may be higher or lower than that on the unfinned surfaces. For example, the interrupted (strip, louver, etc.) fins provide both an increased area and increased heat transfer coefficient, while the internal fins in a tube, meant primarily for structural strength and flow mixing purposes, may result

into a slight reduction in the heat transfer coefficient depending on fin spacing.

Loosely defined, a compact heat exchanger is one which incorporates a heat transfer surface having a high "area density". That is, a high ratio of heat transfer surface to volume. Somewhat arbitrarily, it is specified that a compact surface has an area density β greater than $700 \text{ m}^2/\text{m}^3$. A spectrum of surface area densities of heat exchanger surfaces is shown in Fig. 3. On the bottom of the figure, two scales are shown: the hydraulic diameter D_h in mm and equivalent heat transfer surface area density $\beta (\text{m}^2/\text{m}^3)$. Different exchanger - surfaces are shown in rectangles. The short vertical sides of a rectangle, when projected on the β (or D_h) scale, indicate the range of surface area density (or hydraulic diameter) for the particular surface in question. Interesting details, for heat exchanger classification have been given [16] with adequate figures. The motivation for using compact surfaces is to gain specified heat exchanger performance, $q/\Delta t_m$, within acceptably low mass and box volume constraints. As

$$\frac{q}{\Delta t_m} = U \beta V ,$$

where q is the heat transfer rate and Δt_m - the mean temperature difference.

"COMPACTNESS?" A MATTER OF DEGREE

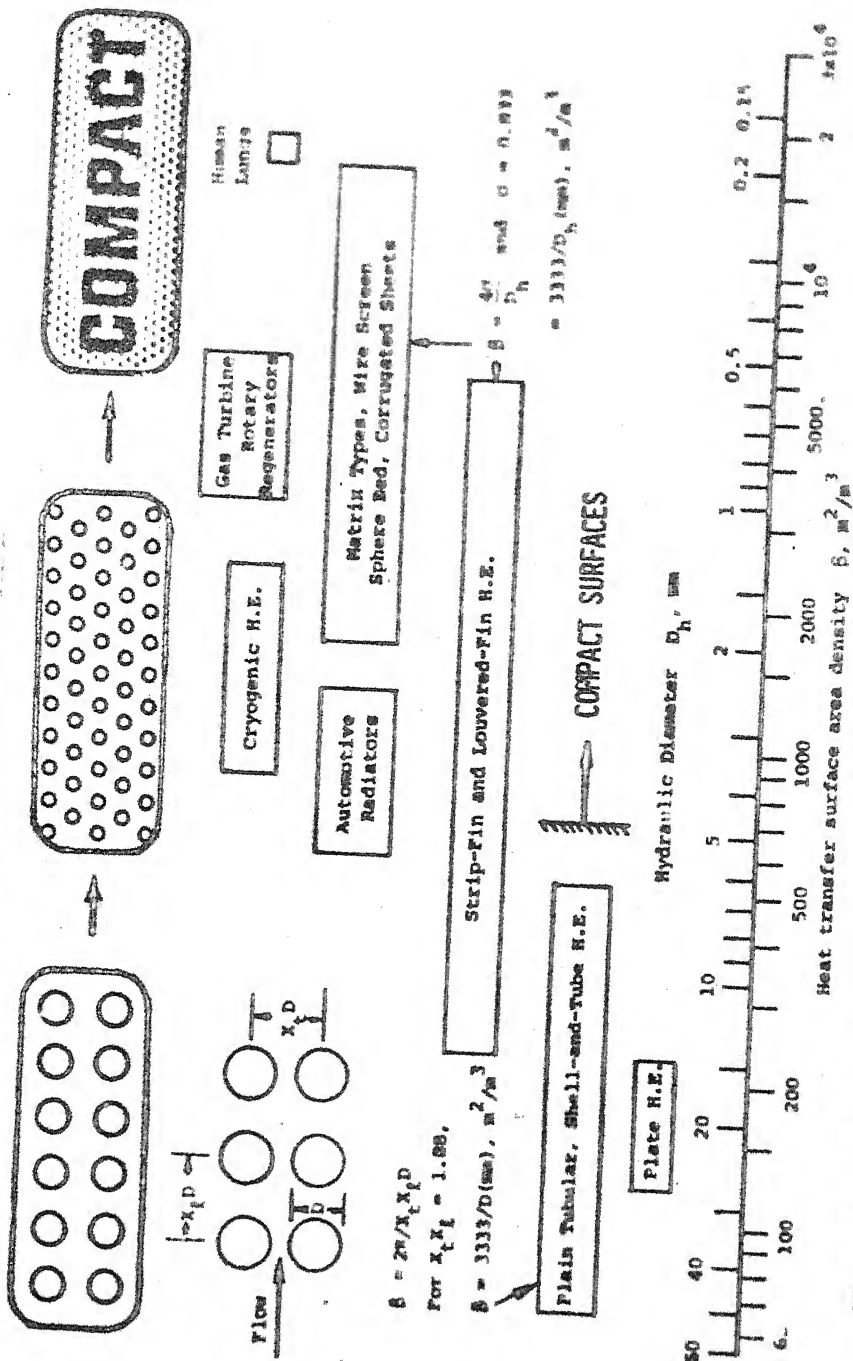


Fig 3. Heattransfer surface area density spectrum of exchanger surfaces

Clearly, a high β minimizes the exchanger volume V . Moreover, compact surfaces generally result in a higher overall conductance U . This also amounts to having a smaller volume. As compact surfaces can achieve structural stability and strength with a thinner section, the gain in lower exchanger mass is even more pronounced than the gain in lower volume. Various techniques employed to make heat transfer surfaces compact are: fins between plates, finned circular tubes, or densely packed continuous or interrupted cylindrical flow passages of various shapes.

A heat exchanger of any structural construction is considered as compact if it employs a compact surface on either one or more sides of a two-fluid or multi-fluid heat exchanger. The convective heat transfer coefficient for gaseous fluids is generally one or two orders of magnitude lower than ^{those of} water, oil, and other fluids. Thus, to reduce the size and weight of a gas-to-liquid heat exchanger, the heat transfer surface on the gas side needs to be much more compact as compared to a simple heat exchanger having only circular tube. Hence, for a somewhat "balanced" design, a compact surface is employed on the gas side. Thus major applications of compact heat exchangers are gas-to-gas, gas-to-liquid, and gas-to-condensing or evaporating fluid heat exchangers.

3.2 CONSTRUCTION TYPES AND SURFACE GEOMETRIES

The basic construction types employed in the design of a compact heat exchanger are extended surface heat exchangers employing fins on one or more sides, regenerators employing small hydraulic diameter surfaces, and tubular exchangers employing small diameter tubes.

Two most common types of extended surface exchangers are the plate-fin and tube-fin types. In a plate-fin exchanger, fins or spacers are sandwiched between parallel plates, referred to as parting plates or parting sheets or formed tubes. Fins are attached to the plates by brazing, soldering, gluing, welding, mechanical fit, or by extrusion. While the plates separate the two fluid streams, the fins form the individual flow passages. Alternate fluid passages are connected in parallel by suitable headers to form two or more sides of the exchanger. The plate-fins are categorized as

1. Plain (uncut surfaces) and straight fins
2. Plain but wavy fins
3. Interrupted fins

Plain triangular and rectangular fins obviously belong to first category. Strip, louver, perforated and pin fins comprise the last category. The heat transfer coefficient

and friction factor in the developing flow region are considerably higher than those in the fully developed region. This is because the developing boundary layers are thinner and offer lower thermal and hydro dynamic resistances compared to those for the thick boundary layers associated with the fully developed flows. Wavy and interrupted fins have boundary layers developing after each interruption. With a proper design, the resultant heat transfer coefficients and heat transfer rates are significantly higher at the same pressure drops for wavy and interrupted fins compared to those for the plain fins. Hence, these fins use the materials very efficiently if the design constraints allow the use of such fins.

In a tube-fin exchanger, tubes of round, rectangular or elliptical shapes are generally used. Fins outside the tubes may be categorized as

1. Normal fins on individual tubes
2. Longitudinal fins on individual tubes
3. Continuous (plain, wavy, interrupted) fins on an array of tubes.

Fins inside the tubes are classified as integral or attached fins.

3.3 HEAT EXCHANGER DESIGN METHODOLOGY

Two most common heat exchanger design problems are the rating and sizing problems. For an existing heat

exchanger, the performance evaluation problem is referred to as the rating problem. To arrive at a design of a new exchanger to meet the specified performance, is referred to as a sizing problem. The objective in rating is either to verify Vendor's specifications or to determine the performance at off-design conditions. In this problem, the following quantities are specified: the exchanger construction type, flow arrangement, overall core dimensions, complete details on the material and surface geometries on both sides including their heat transfer and pressure drop characteristics, fluid flow rates, inlet temperatures, and fouling factors. The designer's task is to predict the fluid outlet temperatures, heat transfer rate, and pressure drop on each side.

Sizing involves a design of a new heat exchanger for specified performance within known constraints and minimum objective function. Here the fluid inlet and outlet temperatures and flow rates are generally specified as well as the pressure drop on each side. The designer selects the construction type, flow arrangement, materials and surfaces on each side to determine the core dimensions to meet the specified heat transfer and pressure drop requirements. If the problem is not well posed without even minimum necessary information, the designer arrives at the necessary information based on the discussions

with the customer, his own experience and engineering judgements.

The package approach suggested by Shah et al. [12] has been adopted in the present work for the heat exchanger design. The complete procedure is given in the flow chart form (Fig. 4.) In this method each possible surface geometry and construction type is considered to be an alternative design. Thus there may be several independent optimized solutions satisfying the problem requirements. Engineering judgement, comparison of objective function values, and other evaluation criteria are then applied to select a final optimum solution for implementation. Thus after arriving at the possible alternate designs, the total number of constraints for each design are formulated. This includes the customer's explicitly specified constraints like fixed frontal area, ranges of heat exchanger dimensions and implicit constraints like minimum heat transfer, and allowable pressure drop. Starting with a set of dimensions for the geometry and input operating conditions, the rating problem is solved. The output from the heat exchanger calculations is fed to the optimization package where the constraints and objective function are evaluated. New values for the design variables are subsequently generated and heat exchanger calculations are repeated. The iterations are continued until the objective function is

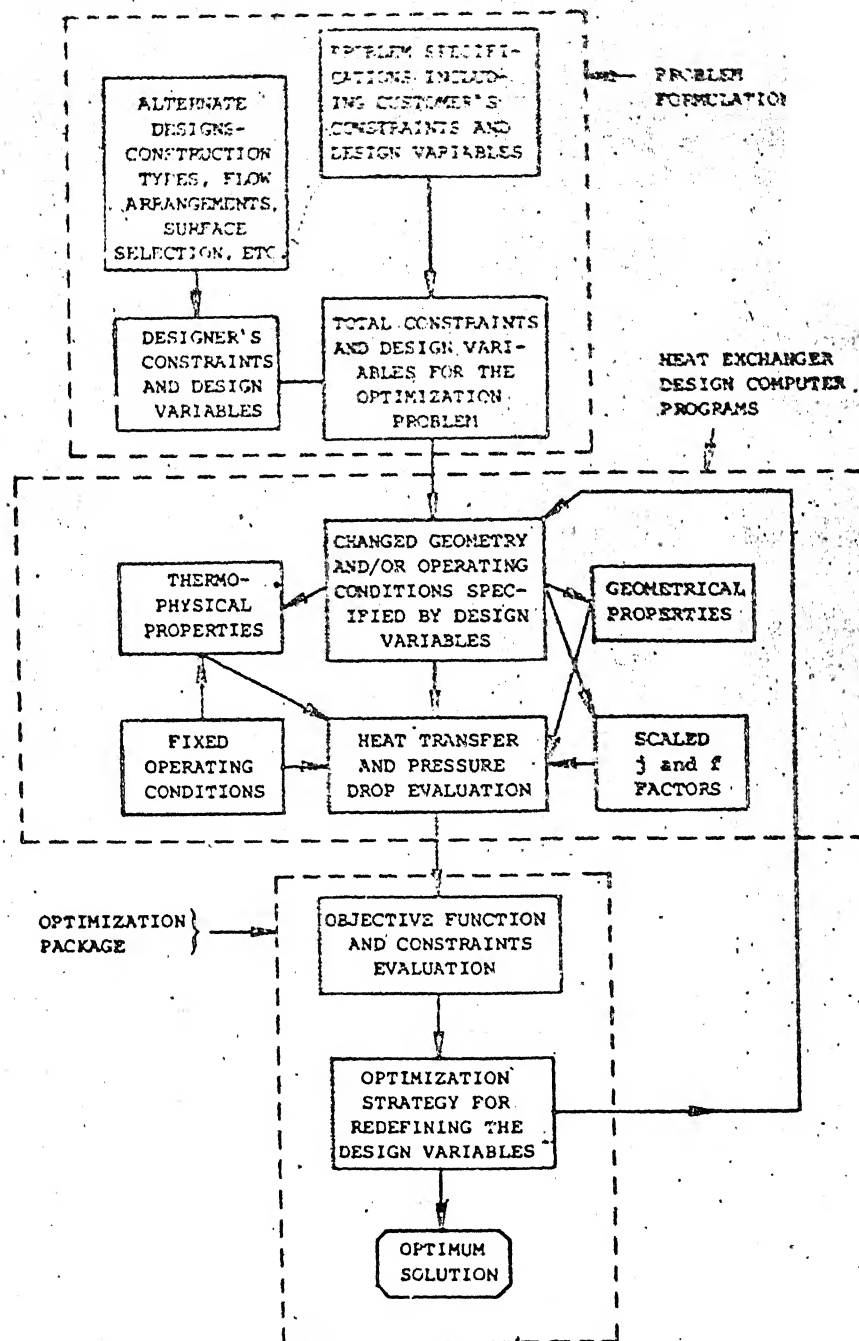


Fig 4. Heat exchanger optimization methodology

optimized within the desired accuracy and all the constraints are satisfied.

A glance at ~~xxx~~ Fig. 4 indicates two computer program packages needed for the optimization are the heat exchanger design programs and the optimization package. It should be emphasized that the heat exchanger programs must be working properly before any attempt is made to implement the optimization package. Otherwise, "garbage-in gospel-out" will result from the "black box".

CHAPTER 4

OPTIMUM DESIGN OF A COMPACT HEAT-EXCHANGER

4.1 INTRODUCTION

A variety of heat-exchanger design-methods have been proposed to determine the best performance of a heat exchanger based on specific performance criteria. However, little work [12, 14] has been done applying the concepts and techniques of non-linear programming to optimizing heat exchanger design. These techniques offer a systematic approach for improving heat exchanger performance by allowing trade-offs to be made between competing objectives such as pressure drop, power, cost, volume and weight. The optimization package developed in Chapter-2 is quite general and may be used for any design problem. The present chapter is devoted to test the applicability and efficacy of the general package for the design of a compact heat exchanger for a gas-turbine plant. Relevant data for the design parameters have been taken from Appendix-B of Kays and London [15]. The objective is to minimize pumping power requirements with constraints on the exchanger size, volume and the total heat transferred. The design variables are the hot and cold fluid capacity

flows and linear dimensions of the heat exchanger. The ϵ - NTU approach is used to evaluate the heat exchanger performance. The design procedure outlined in this chapter is quite general and suitable for any heat exchanger application.

4.2 NOMENCLATURE

A_a	Exchanger total heat transfer area on air side
A_c	Free flow area
A_f	Exchanger total fin area on air side
A_{fra}	Exchanger total frontal area on air side
A_{frw}	Exchanger total frontal area on water side
B	Exchanger breadth (Fig. 5)
C_a	Flow-stream capacity rate of air
C_w	Flow-stream capacity rate of water
C_{min}	Minimum of C_a or C_w
C_{max}	Maximum of C_a or C_w
C_{pa}	Specific heat at constant pressure for air
C_{pw}	Specific heat at constant pressure for water
f_a	Friction factor on air side
f_w	Friction factor on water side
G_a	Air stream mass velocity
G_w	Water stream mass velocity
g_c	Proportionality factor in Newton's second law
h_a	Heat transfer film coefficient on air side

h_w	Heat transfer film coefficient on water side
H	Exchanger height (Fig. 5)
k_a	Thermal conductivity of air
k_w	Thermal conductivity of water
k_f	Thermal conductivity of fin material
L	Exchanger fluid-flow length
l_f	Fin length
M	Molecular weight of air
m	A fin effectiveness parameter
P_a	Pressure on air side
P_w	Pressure on water side
Q	Total heat transferred
Q_D	Total desired heat transfer
R	Universal gas constant
C^*	Capacity ratio C_{\min}/C_{\max}
r_a	Hydraulic radius on air side
r_w	Hydraulic radius on water side
T_c	Temperature of cold fluid (water)
T_h	Temperature of hot fluid (air)
U	Overall thermal conductance
V	Volume of heat exchanger
v	Specific volume of air
W	Exchanger width (Fig. 5)

Greek Symbols

α	Ratio of total transfer area on one side of the exchanger to total volume of the exchanger
δ_f	Fin thickness
ε	Exchanger effectiveness
η_f	Fin temperature effectiveness
η	Total surface temperature effectiveness
σ	Ratio of free flow area to frontal area
μ	Viscosity coefficient
ρ	Density

Dimensionless Groupings

Re	Reynolds number ($4 r G/\mu$)
Pr	Prandtl number ($\mu C_p/k$)
NTU	Number of transfer units (AU/C_{\min})
j	Colburn factor ($h P_r^{2/3} / G C_p$)

Subscripts

a	Air side
avg	Average
i	Inlet conditions
o	Outlet conditions

4.3 PROBLEM FORMULATION

4.3.1 Design Problem

The heat exchanger configuration is shown in Fig.5. The inlet pressure, temperature and flow stream capacity rate of the hot air are T_{hi} (126.7°C), P_a (275.8 kPa) and C_a respectively. The values in the parenthesis have been (A GAS TURBINE HEAT EXCHANGER) chosen for a specific application. The hot air is cooled by water, pumped at the rate of C_w and at ambient temperature of T_{ci} (15.6°C) and pressure P_w (101.04 kPa). It is required to minimize the total fluid pumping power for this exchanger by varying the heat capacity rates C_a , C_w on each side and also varying the height, width and breadth of the exchanger. It should be spaced within a volume of V_{\max} and it has a heat transfer rate of atleast Q_D .

4.3.2 Selection Aspects

First of all, a cross flow arrangement with both fluids unmixed is selected. For a complete optimization, every possible combination of available heat transfer surfaces should be considered in order to make, reasonably, sure, that the optimum design is obtained. This may be impractical and ~~too~~ time consuming. For the given pressure, temperature ranges and space constraint, as selected for the present work in 4.3.1, a finned-flat tube core configuration is suitable [15]. Once the basic surface geometry

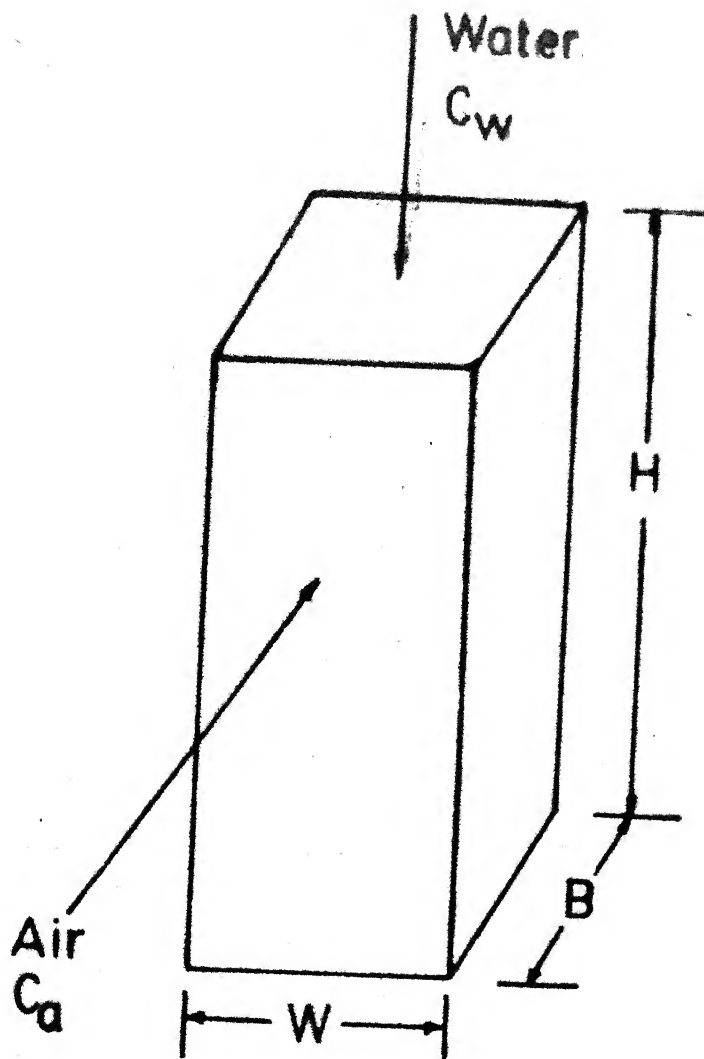


Fig 5. Heat exchanger Configuration
&
Design variables

is selected, the designer gets the values of various geometric parameters, as listed below, that enter the heat transfer calculation.

1. Hydraulic radius, r
2. Heat transfer area/Total volume, α
3. Free flow area/Frontal area, σ
4. Fin area/Heat transfer area, A_f/A
5. Fin thickness, δ_f
6. Fin conductivity, k_f
7. Fin length, l_f .

These are to be determined for the hot and cold fluid sides of the heat exchanger. The heat core selected in this application uses fins only on the air side for the obvious reason that the heat transfer film coefficient is lower on this side. The friction and colburn factors for the flow inside the flattened tubes are obtained from [15] assuming the ratio $L/4r_h = 40$, and are given in figures 6 and 7.

4.3.3 Optimization Problem Definition

The design variables are exchanger width W , height H , breadth B , and the capacities C_a and C_w . By appropriately scaling and nondimensionalizing these variables, a new set of parameters may be defined which have the same order of magnitude. This makes the operation of optimization procedure more convenient. Defining

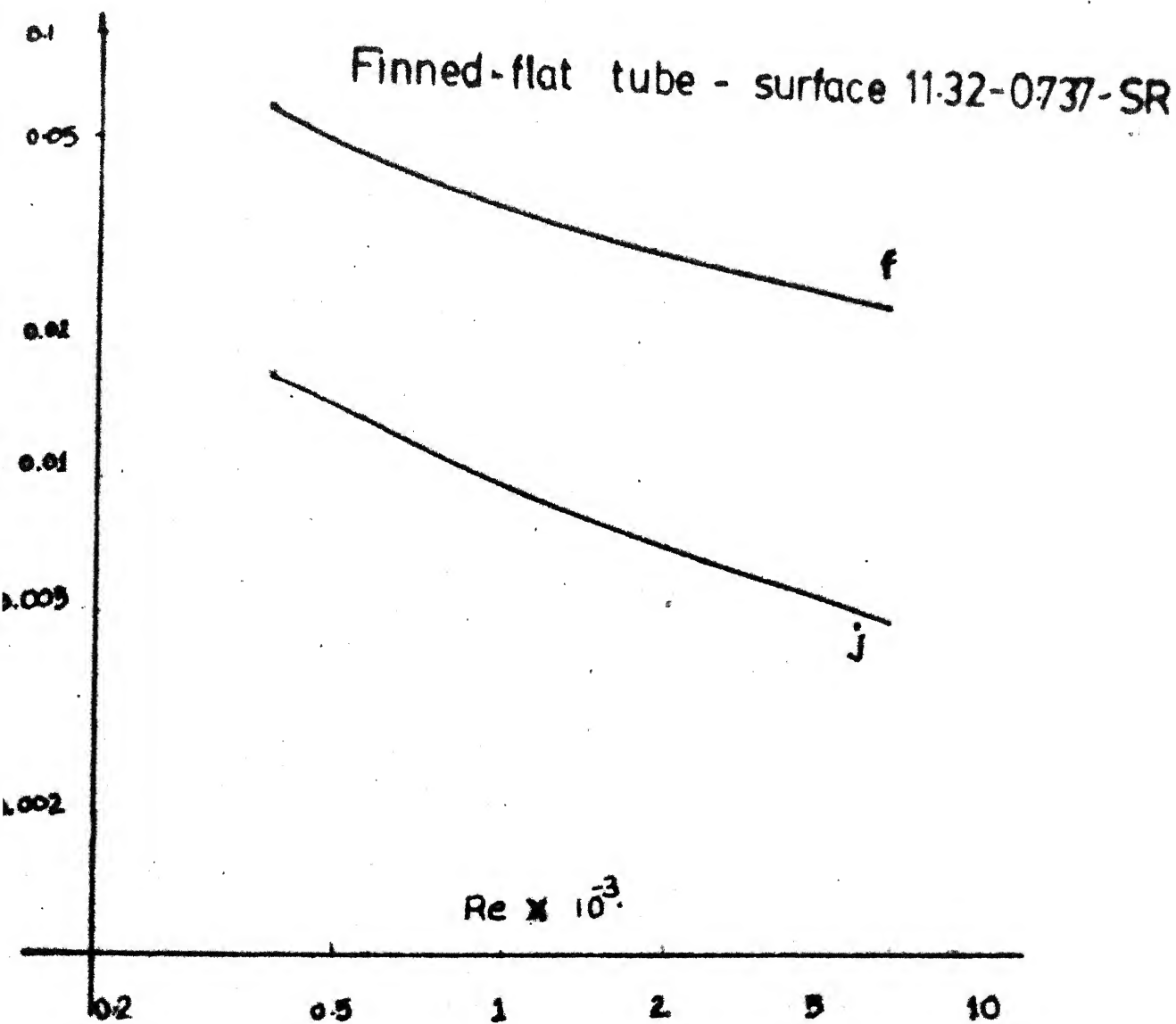


Fig 6. Friction & Colburn factors for Air side

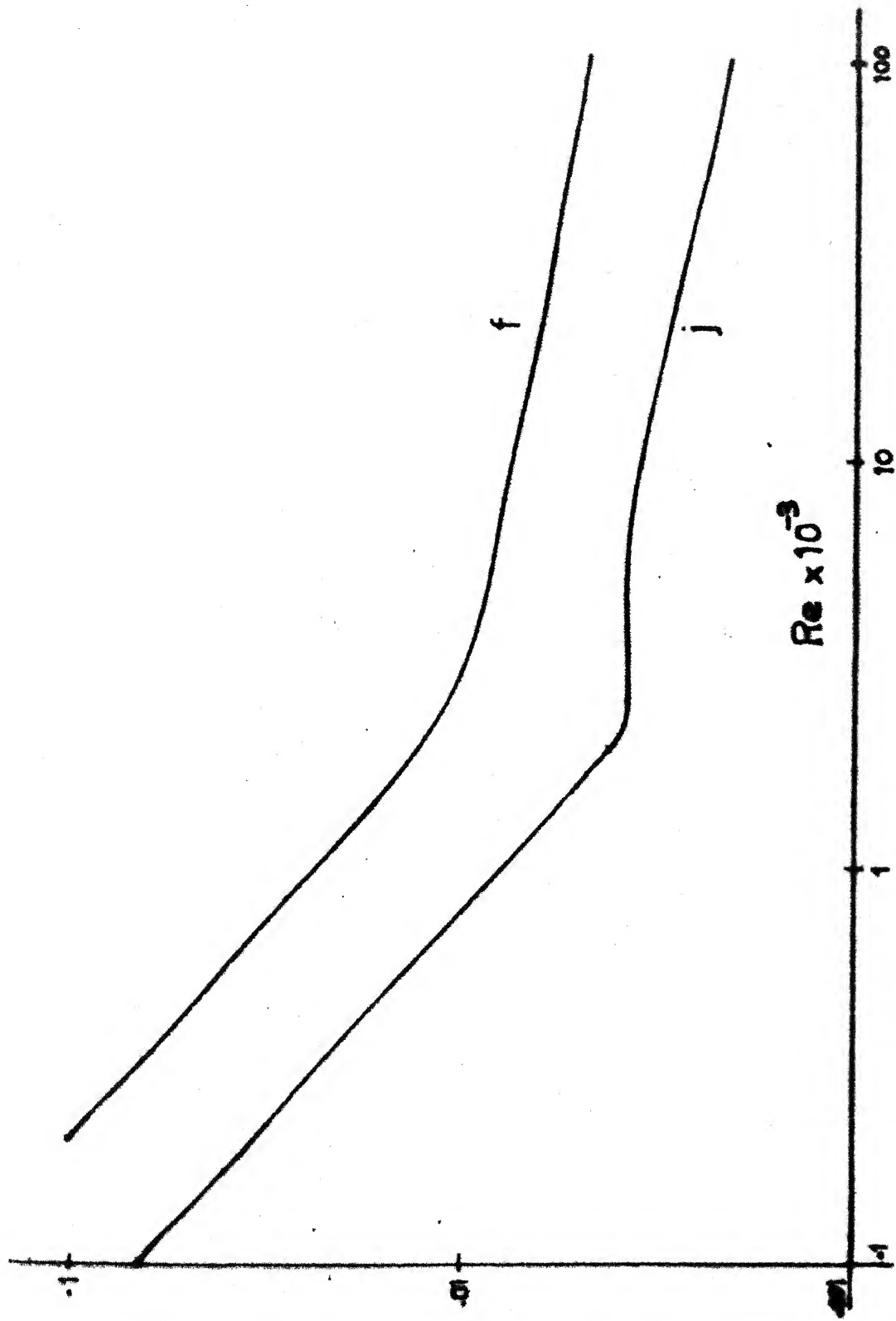


Fig 7. Friction & Colburn factors for Tube side

$$L = (V_{\max})^{1/3} \quad (11)$$

$$Z = \frac{Q_D}{T_{hi} - T_{ci}}$$

the variables W , H , B , C_a and C_w are nondimensionalized and the design vector is defined as follows:

$$\begin{aligned} X_1 &= C_a/Z \\ X_2 &= C_w/10Z \\ X_3 &= W/L \\ X_4 &= B/L \\ X_5 &= H/L \end{aligned} \quad (12)$$

Representing the pumping power as ϕ , the minimization problem is:

$$\text{Minimize } \phi(\bar{X}) \quad (13)$$

Subject to the following constraints.

- i) To maintain the minimum value of the heat transfer rate, $Q \geq Q_D$

$$\therefore g_1 = 1 - \frac{Q}{Q_D} \leq 0$$

- ii) To space the exchanger within a certain volume

$$V \leq V_{\max}$$

$$\therefore g_2 = \frac{V}{V_{\max}} - 1 \leq 0$$

- iii) Realizing the need for a meaningful configuration, the following constraints are imposed:

Width should be less than twice the breadth, and
Height should be less than thrice the width
i.e.

$$x_3 \leq 2 x_4$$

$$x_5 \leq 3 x_3$$

$$g_3 = \frac{x_3}{2x_4} - 1 \leq 0$$

$$g_4 = \frac{x_5}{3x_3} - 1 \leq 0$$

- iv) Upper bounds for the capacities are given some practical values say BIG

$$C_a < \text{BIG}$$

$$C_w < 10 \text{ BIG}$$

The water capacity rate is taken to be of one order higher than air capacity rate.

$$g_5 = \frac{x_1}{\text{BIG}} - 1 \leq 0$$

$$g_6 = \frac{x_2}{10 \text{ BIG}} - 1 \leq 0$$

- v) Upper bound for the remaining variable breadth is given as $1/L$ so as to satisfy the volume constraint always:

$$g_7 = L x_4 - 1 \leq 0$$

vi) Non-negativity constraints are finally imposed on all the five variables.

$$g_8 = -x_1 < 0$$

$$g_9 = -x_2 < 0$$

$$g_{10} = -x_3 < 0$$

$$g_{11} = -x_4 < 0$$

$$g_{12} = -x_5 < 0$$

4.4 PERFORMANCE CALCULATION PROCEDURE

4.4.1 Overview

The heat exchanger performance is evaluated in an iterative manner as shown in Fig. 8. An iterative scheme has to be used since the outlet fluid temperatures and heat exchanger effectiveness are not known at the outset. A value of 0.75 is assumed for the effectiveness ϵ at the start of the iteration scheme. The outlet fluid temperatures can then be found. The fluid properties vary with temperature and are evaluated at the mean of the inlet and outlet temperatures. The fluid properties appear in the expressions for mass velocities (G), and Reynolds (Re) and Prandtl (Pr) numbers. The Colburn 'j' and friction 'f' factors are function of Re. With j and Pr known, the film coefficient h can be computed, then the overall conductance U, and finally the number of transfer units NTU. Exchanger effectiveness ϵ is a

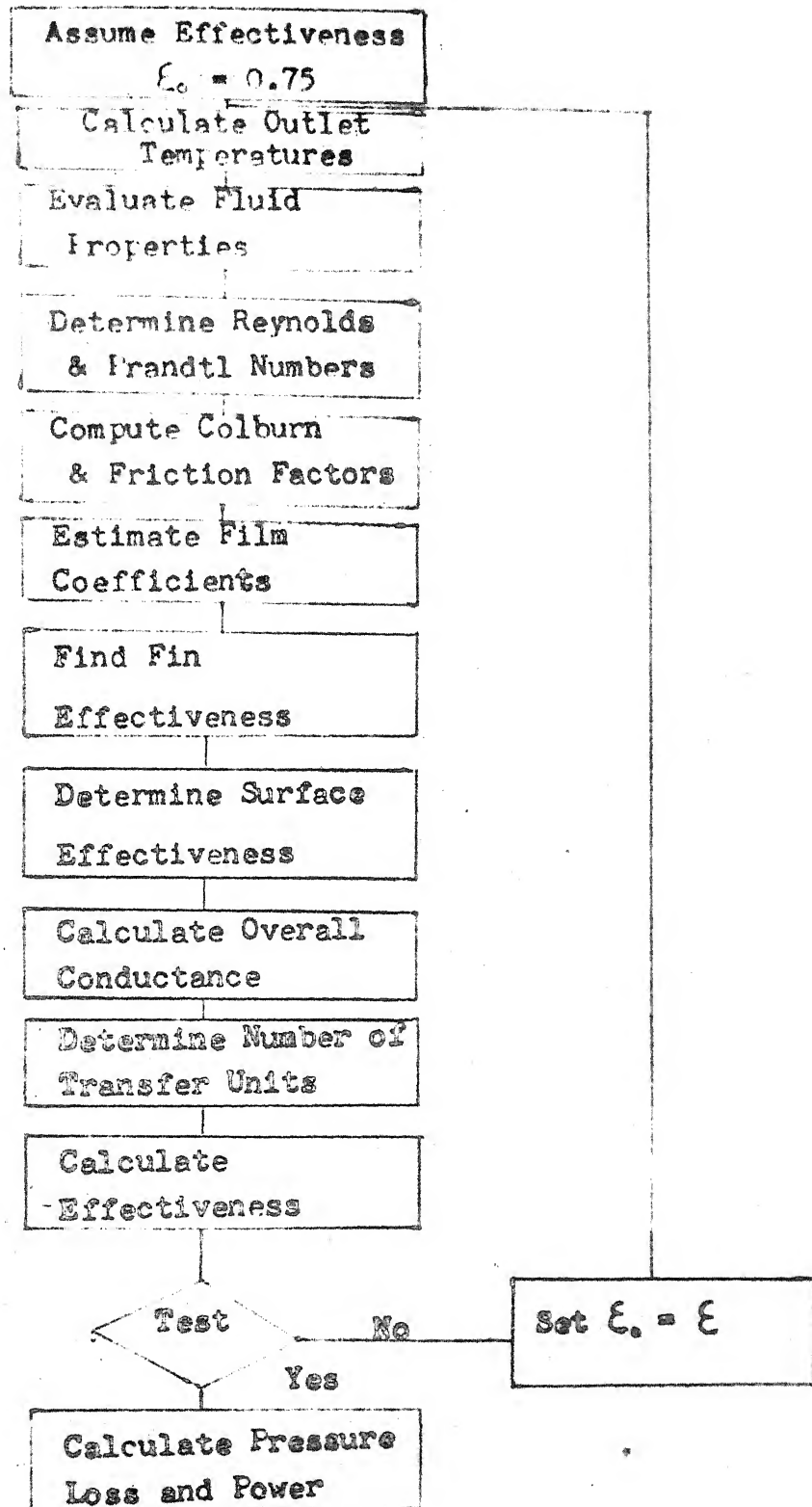


Fig 8 . PERFORMANCE ANALYSIS CHART

function of NTU and capacity ratio C^* . Expressions for ϵ are available for different flow configurations. The computed value of ϵ is then compared with the assumed value for concurrency.

4.4.2 Core Dimensions

The geometrical parameters for the selected finned-flat tube surface 11.32-0.737-SR [15] are:

i) Air side:

$$\begin{aligned}
 r_a &= 0.00288 \\
 \alpha_a &= 270 \\
 A_f/A_a &= 0.845 \\
 \sigma_a &= 0.78 \\
 \delta_f &= 0.00033 \\
 k_f &= 100 \\
 l_f &= 0.01875
 \end{aligned}
 \tag{14}$$

ii) Water side:

$$\begin{aligned}
 r_w &= 3.06 \times 10^{-3} \\
 \alpha_w &= 42.1 \\
 \sigma_w &= 0.129
 \end{aligned}
 \tag{15}$$

From Fig. 5, the frontal areas for the air and water, sides of the intercooler can be readily computed. The frontal area is the cross-sectional area normal to the direction

of the flow.

$$\begin{aligned}
 A_{fra} &= WH \\
 &= X_3 X_5 L^2 \\
 A_{frw} &= WB \\
 &= X_3 X_4 L^2
 \end{aligned}
 \tag{16}$$

The total volume of the heat exchanger is :

$$\begin{aligned}
 V &= WHB \\
 &= X_3 X_4 X_5 L^3
 \end{aligned}
 \tag{17}$$

4.4.3 Outlet Fluid Temperatures

From equation(12) we have

$$\begin{aligned}
 C_a &= X_1 Z \\
 C_w &= 10 X_2 Z
 \end{aligned}
 \tag{18}$$

Defining C_{min} to be the minimum of C_a and C_w , the maximum possible amount of heat that can be transferred by the heat exchanger is given by:

$$Q_{max} = C_{min} (T_{hi} - T_{ci}) \tag{19}$$

If the intercooler effectiveness is ϵ , then the actual amount of heat exchanged between the hot and cold fluid streams is:

$$\begin{aligned}
 Q &= \epsilon Q_{max} \\
 &= \epsilon C_{min} (T_{hi} - T_{ci})
 \end{aligned}
 \tag{20}$$

The heat balance equations are:

$$\begin{aligned} Q &= C_a (T_{hi} - T_{ho}) \\ &= C_w (T_{co} - T_{ci}) \end{aligned} \quad (21)$$

Substituting (20) in (21), the outlet temperatures are:

$$\begin{aligned} T_{ho} &= T_{hi} - \frac{\varepsilon C_{min} (T_{hi} - T_{ci})}{C_a} \\ T_{co} &= T_{ci} + \frac{\varepsilon C_{min} (T_{hi} - T_{ci})}{C_w} \end{aligned} \quad (22)$$

The optimizer will manipulate the design variables in order to find optimum, ensuring at the same time that the above energy balance is satisfied.

4.4.4 Fluid Properties

For heat exchanger applications, it is a normal practice to estimate the transport properties, which are temperature dependent, at the bulk temperature which is the average of the fluid inlet and outlet temperatures. The bulk temperatures for the air and water streams are:

$$\begin{aligned} T_{cavg} &= \frac{T_{co} + T_{ci}}{2} \\ T_{havg} &= \frac{T_{ho} + T_{hi}}{2} \end{aligned} \quad (23)$$

Experimental data for the fluid properties of air and water at different temperatures are readily available[19].

In this optimization study, polynomial expressions are fitted to the experimental data using the least squares technique (Fig. 9). The perfect gas equation, $Pv=RT/M$, is used to determine the specific volume of air at the inlet, outlet, and bulk temperatures with the pressures assumed to remain constant at P_a .

$$\begin{aligned}
 v_i &= \frac{53.345 (460 + T_{hi})}{133 P_a} \\
 v_o &= \frac{53.345 (460 + T_{ho})}{144 P_a} \\
 v_{avg} &= \frac{53.345 (460 + T_{havg})}{144 P_a}
 \end{aligned} \tag{24}$$

4.4.5 Reynolds and Prandtl Number

The mass flow is given by dividing the heat capacity C by the specific heat C_p . With the free flow area known, the mass velocities for air and water are:

$$\begin{aligned}
 G_a &= \frac{C_a}{C_{pa} A_{ca}} = \frac{C_a}{C_{pa} \sigma_a A_{fra}} \\
 G_w &= \frac{C_w}{C_{pw} A_{cw}} = \frac{C_w}{C_{pw} \sigma_w A_{frw}}
 \end{aligned} \tag{25}$$

The Reynolds and Prandtl numbers can now be computed as:

$$\begin{aligned}
 Re_a &= \frac{4 \cdot G_a \cdot r_a}{\mu_a} \\
 Re_w &= \frac{4 \cdot G_w \cdot r_w}{\mu_w} \\
 Pr_a &= \frac{\mu_a C_{pa}}{k_a} \\
 Pr_w &= \frac{\mu_w C_{pw}}{k_w}
 \end{aligned}
 \tag{26}$$

4.4.6 Friction and Colburn Factors

Polynomial expressions, Fig. 9 are fitted to the experimental data as shown in Figures 6 and 7 by least square method. In practice, the experimental data is approximated by several polynomials which are valid for a small range of Reynolds number. Since most compact heat exchangers operate in the transition flow regime ($100 < Re < 10,000$) it is important to get good polynomial fits for f and j in this flow regime.

4.4.7 Heat Transfer Film Coefficient

Once the colburn factors for the hot and cold fluids are known, the film coefficients are calculated from the following equations:

$$\begin{aligned}
 h_a &= \frac{j_a G_a C_{pa}}{(Pr_a)^{2/3}} \\
 h_w &= \frac{j_w G_w C_{pw}}{(Pr_w)^{2/3}}
 \end{aligned}
 \tag{27}$$

Expression for Fluid Properties :

Water

$$\text{Let } G = \frac{T_{\text{cavg}}}{100}$$

$$C_{pw} = .00178 G^4 - .01789 G^3 + .06763 G^2 - .09169 G + 1.0348$$

$$\rho_w = 10 (-.00193 G^4 + .02180 G^3 - .1300 G^2 + .08017 G + 6.2283)$$

$$\mu_w = .0547 G^4 - .56414 G^3 + 2.6107 G^2 - 3.5424 G + 5.1444$$

$$R_w = .1 (-.00332 G^4 - .04371 G^3 - .295151 G^2 + .88891 G + 3.0051)$$

Air

$$\text{Let } G = \frac{T_{\text{havg}}}{1000}$$

$$C_{pa} = .1 (.03307 G^4 - .19583 G^3 + .30682 G^2 + .20886 G + 2.3653)$$

$$\mu_a = .1 (-.00571 G^4 + .05756 G^3 - .22801 G^2 + .67541 G + .40343)$$

$$R_a = .01 (.00735 G^4 - .04446 G^3 - .10931 G^2 + 2.0579 G + 1.4882)$$

Friction and Colburn Factors :

Valid for 1000 Re 10,000

Air

$$\text{Let } Y = \log_{10} (Re_a \cdot 10^{-3})$$

$$\log_{10}(100f_a) = .11998 Y^4 - .04463 Y^3 + .26677 Y^2 - .45965 Y + .58784$$

$$\log_{10}(100j_a) = -.11873 Y^4 + .03586 Y^3 + .22707 Y^2 - .4789 Y - .00622$$

Water

$$\text{Let } Y = \log_{10} (Re_w \cdot 10^{-3})$$

$$\log_{10}(100f_w) = -1.2239 Y^4 + 2.1643 Y^3 - .56969 Y^2 - .87268 Y + .38629$$

$$\log_{10}(100j_w) = -1.3431 Y^4 + 1.6032 Y^3 + .5549 Y^2 - 1.2064 Y - .08379$$

4.4.8 Fin Effectiveness

For finned-tube compact surfaces, the fins are assumed to be straight uniform fins. The fin effectiveness for this type of fin is shown to be [20].

$$\eta_f = \frac{\tanh (ml_f)}{ml_f} \quad (28)$$

where

$$m = \left(\frac{2h_a}{k_f \delta_f} \right)^{1/2}$$

The surface effectiveness of the heat transfer surface on the air side is given by [20]:

$$\eta = 1 - \frac{A_f}{A_a} (1 - \eta_f) \quad (29)$$

4.4.9 Exchanger Effectiveness

The conductance of the heat transfer surface can be computed from the film coefficients and the surface effectiveness. In computing the overall conductance, the wall resistance of the surface is neglected.

$$\frac{1}{U_a} = \frac{1}{\eta h_a} + \frac{1}{\frac{\alpha_w}{\alpha_a} h_w} \quad (30)$$

Equation (30) gives the overall conductance based on the air side heat transfer area. The number of transfer units (NTU) is

$$NTU = \frac{A_a U_a}{C_{\min}} = \frac{\alpha_a V U_a}{C_{\min}} \quad (31)$$

Let C_{\max} represent the maximum of C_a and C_w . Then the capacity ratio C^* is:

$$C^* = \frac{C_{\min}}{C_{\max}} \quad (32)$$

With NTU and C^* known, the effectiveness ε for the cross-flow intercooler with both fluids unmixed, is determined from the series solution of Mason [21]

$$\varepsilon = \frac{1}{(NTU)C^*} \sum_{n=0}^{\infty} \left[1 - e^{-NTU} \sum_{m=0}^n \frac{(NTU)^m}{m} \right. \\ \left. 1 - e^{-(NTU)C^*} \sum_{m=0}^{\infty} \frac{[(NTU)C^*]^m}{m} \right] \quad (33)$$

This series is rapidly convergent. The value of ε from (33) is compared with the value assumed at the beginning of the iteration. If these two values differ by more than 0.01, the performance calculations are repeated, using the new value ε for the next iteration. A few iterations are needed to obtain agreement between the assumed and computed effectiveness of the compact heat exchanger.

4.4.10 Pressure Drops

The pressure loss in the fluid streams is contributed by ~~the~~ change in fluid momentum, resulting from the changes in the cross-sectional area at inlet and exit, and by viscous friction. The pressure loss for

the air stream is given by [15]

$$\frac{\Delta P_a}{P_a} = \frac{\left(\frac{G_a}{3600}\right)^2 v_i}{2g_c P_a} \left[(1 + \sigma_a^2) \left(\frac{v_o}{v_i} - 1\right) + f_a \frac{A_a}{A_c} \frac{v_{avg}}{v_i} \right] \quad \dots \quad (34)$$

From the definition of hydraulic diameter:

$$\frac{A_a}{A_c} = \frac{B}{r_a} = \frac{X_4 L}{r_a} \quad (35)$$

On substituting (35) in (34), we get

$$\frac{\Delta P_a}{P_a} = \frac{\left(\frac{G_a}{3600}\right)^2 v_i}{2g_c P_a} \left[(1 + \sigma_a^2) \left(\frac{v_o}{v_i} - 1\right) + \frac{f_a X_4 L}{r_a} \frac{v_{avg}}{v_i} \right] \quad \dots \quad (36)$$

The pressure loss associated with the water stream is mainly due to surface friction and is given by:

$$\begin{aligned} \Delta P_w &= \left(\frac{G_w}{3600}\right)^2 \frac{f_w H}{2g_c \rho r_w} \\ &= \left(\frac{G_w}{3600}\right)^2 \frac{f_w X_5 L}{2g_c \rho r_w} \end{aligned} \quad (37)$$

where, g_c is the gravitational constant, and ρ is the average mass density of water.

The pumping power for each media is the product of pressure loss and volume flow rate, and the total power is given by:

$$\phi = \Delta P_a \cdot \frac{C_a \cdot v_{avg}}{C_{pa}} + \frac{\Delta P_w C_w}{\rho C_{pw}} \quad (38)$$

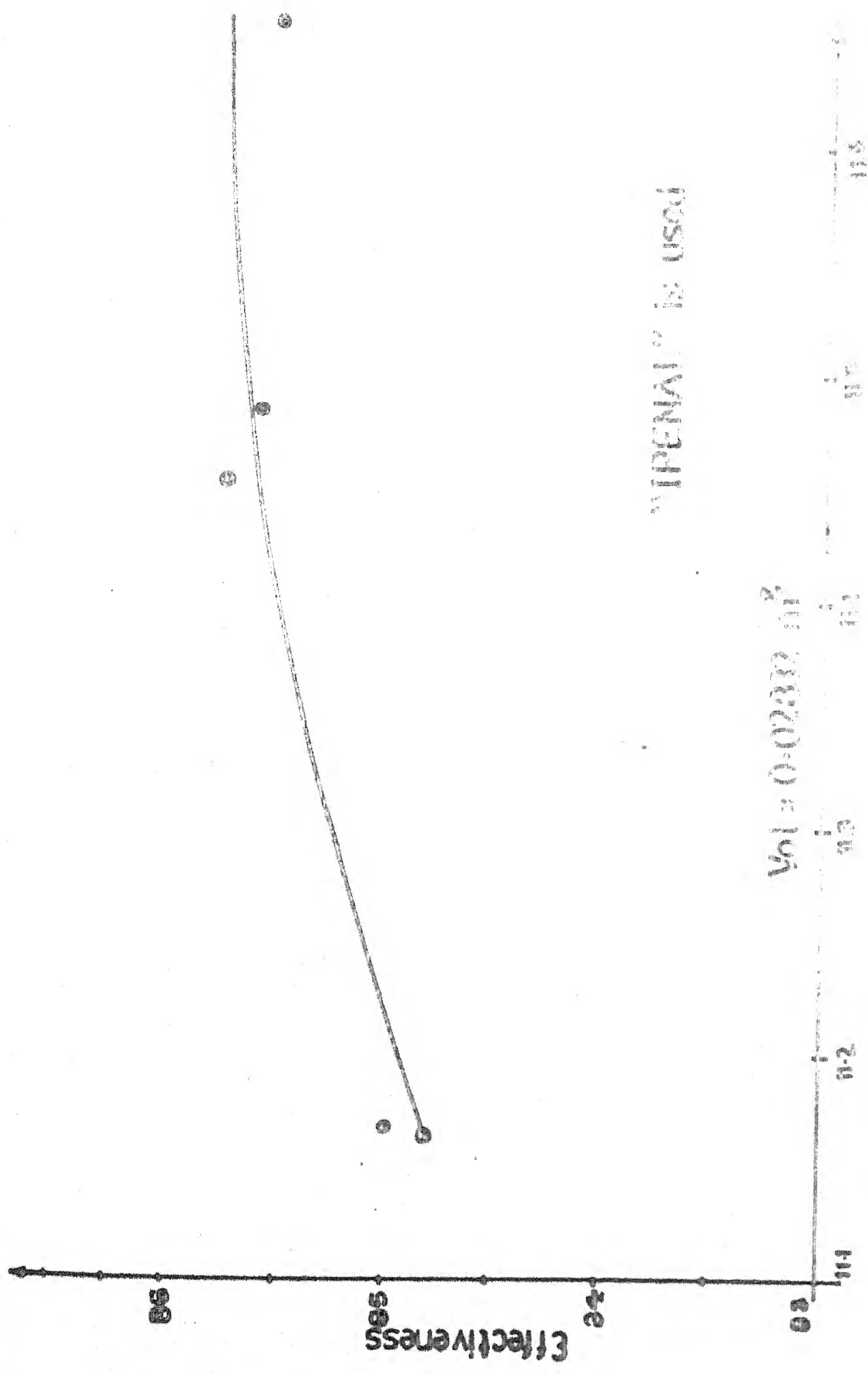
4.5 SOLUTION TECHNIQUES

Three different constrained minimization techniques were applied to minimize the pumping power (Eqn. 13). They are Interior penalty function, Exterior penalty function and the Method of feasible directions. For penalty function methods, the multidimensional minimization schemes tried were Steepest descent, Powell's conjugate directions, Fletcher and Reeves' conjugate gradient and the Davidon-Fletcher-Powell variable metric method. For obvious reasons, Univariate method was ignored. These methods were made to select the Quadratic interpolation procedure for the unidimensional minimization. A graphics program, using the GPGS (General Purpose Graphics System) routines, displays on the graphics terminal the heat exchanger configuration according to the optimized dimensions. This may be used also to give the initial solution for linear dimensions of the exchanger, at the start of the execution of the optimizer, to suit the designer's taste for configuration.

The heat exchanger performance analysis is put in a separate routine, Appendix-A. Finite difference schemes were utilized to get the gradients for various search techniques in the optimizer. All the input parameters used for the heat exchanger performance analysis have been defined in sections 4.2 and 4.3.

A sample of the input data, for the optimizer, used to define the variables and initialize the flags is given here. Results are given in the graphical form. Variation of pumping power and effectiveness with the volume, has been plotted in figures 10 and 11, and a detailed discussion is reserved for the next chapter.

List of Input Data		Significance
NN	= 5	Number of design variables
NC	= 12	Total number of constraints
NI	= 12	Number of inequality constraints
X(1)	= 2.5	Starting point (Feasibility
X(2)	= 1.0	is a must for Interior penalty
X(3)	= 0.5	and Feasible direction
X(4)	= 0.3	methods)
X(5)	= 1.25	
VMAX	= 16.0	Max. volume allowed in a specific run
BIG	= 5.0	Max. capacity rate of both fluids
TT	= 0.1	Initial step size
TMAX	= 0.5	Max. step size allowed
LIM	= 15	Max. number of iterations permitted
FF	= 0.01	Percentage change in variables for FDM computation of gradients
ACC	= 1.E-04	Accuracy desired
R	= 10	Initial, penalty factor



"REPAIR" IS USED

$$Vol = 0.0233 \text{ m}^3$$

Fig 10. Variation of Effectiveness with Volume

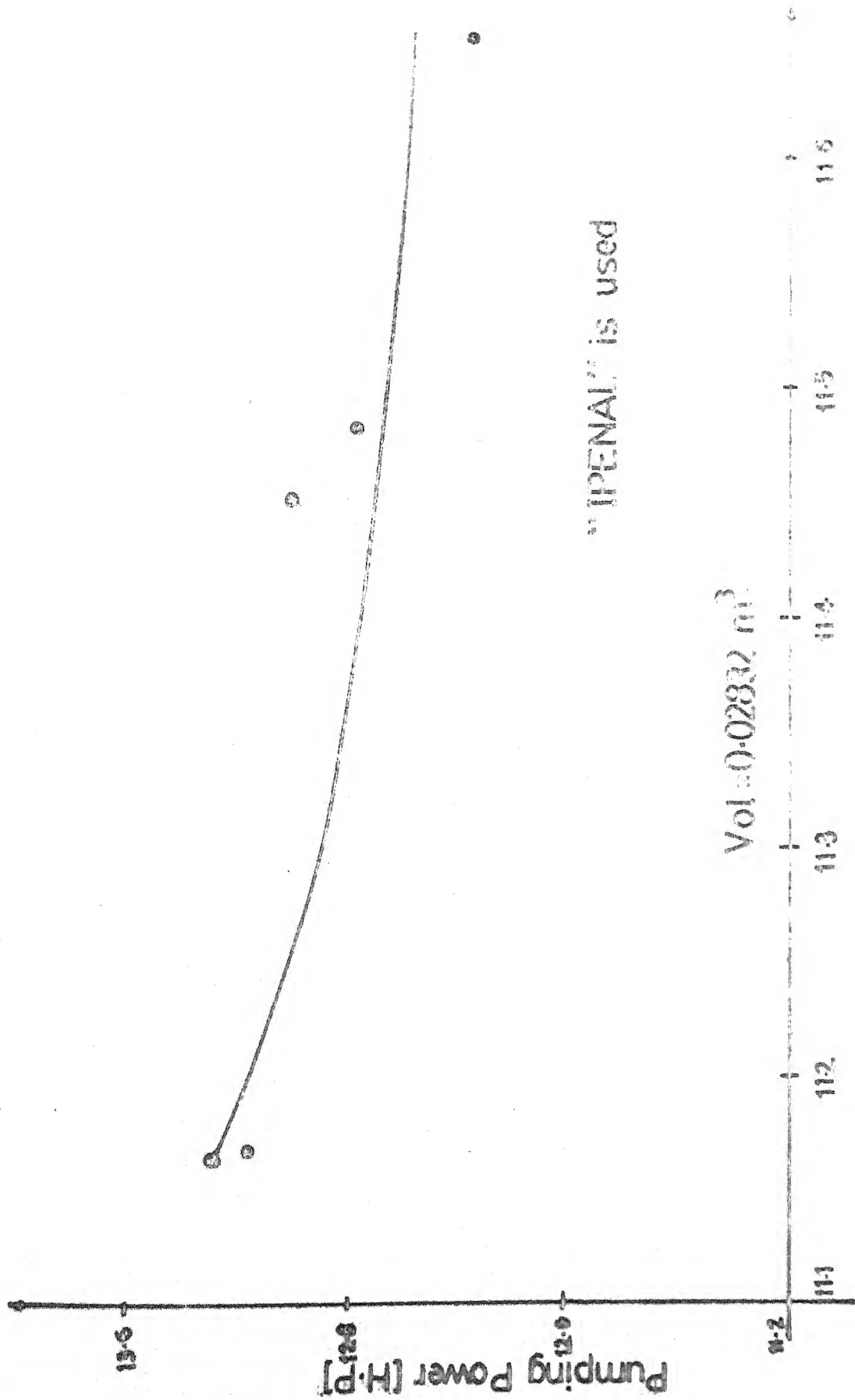


Fig 11. \mathcal{A} vs Vol_{opt}

FC	=	0.1	Factor decreasing R for IPENAL
	=	10	Factor increasing R for EXPEN
W(1:NC)=		0	Push off factors set to '0' for linear constraints
NPERT	=	1	Skips the perturbation scheme

CHAPTER - 5

RESULTS AND DISCUSSION

5.1 TRADE-OFF STUDIES

The compact heat exchanger optimized in the last chapter has the same configuration as considered by Kays and London in example-1 [15]. The objective of the study is to find what reductions in pumping power or volume is possible for this design, while still keeping the same heat transfer performance. The heat exchanger is designed by varying it's dimensions and the mass flow rates of the two fluids viz, air and water. The results, as given in figures 10 and 11, are indicative of the characteristics of a single optimization. Fig. 10 indicates the variation of pumping power with the exchanger volume and Fig. 11 is drawn for effectiveness of the exchanger versus volume. Each point on the curve in Fig. 10 corresponds to the optimum value of the pumping power for a specific size of the exchanger. Similarly, in Fig. 11, the optimum performance of the exchanger is indicated for a particular volume.

However, an expanded optimization study can easily be done by changing the appropriate constraints.

For example, if one is interested in finding the fluid pumping power requirement as a function of the exchanger total volume, V_{\max} is varied and the resultant optimization problem is solved again by the methods used in the last chapter. Optimization results of eqns.(13) for different values of V_{\max} showed that the constraints on linear dimensions of the exchanger are always satisfied and hence, x_3 , x_4 and x_5 are determined by solving these constraint equation as:

$$x_3 = 0.8736$$

$$x_4 = 0.4368$$

$$x_5 = 2.6207$$

The minimization problem in (13) is now reduced to a 2 variables problem and can be restated as

$$\text{Minimize } \phi(x_1, x_2)$$

$$\bar{x}$$

subject to

$$g_1(x_1, x_2) = 1 - \frac{Q(x_1, x_2)}{Q_D}$$

This two-variables problem is solved by the same three different methods of constrained minimization for different values of V_{\max} . A trade-off curve between pumping power and exchanger volume is presented in Fig. 12, variation of effectiveness with volume in Fig. 13, and variation of mass flow rates with the volume is shown in Fig. 14.

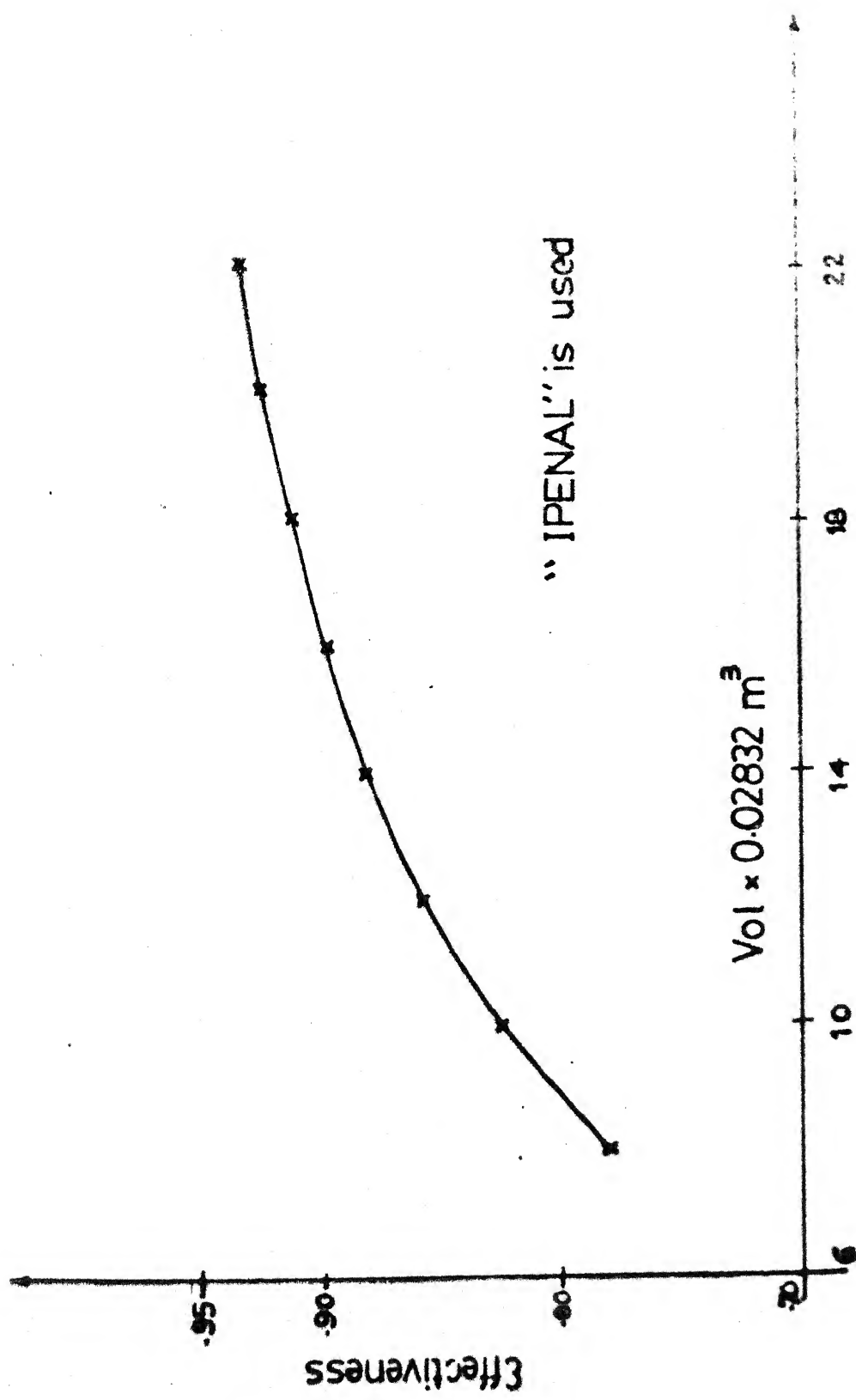


Fig13. Variation of "ε" with Allowable Volume

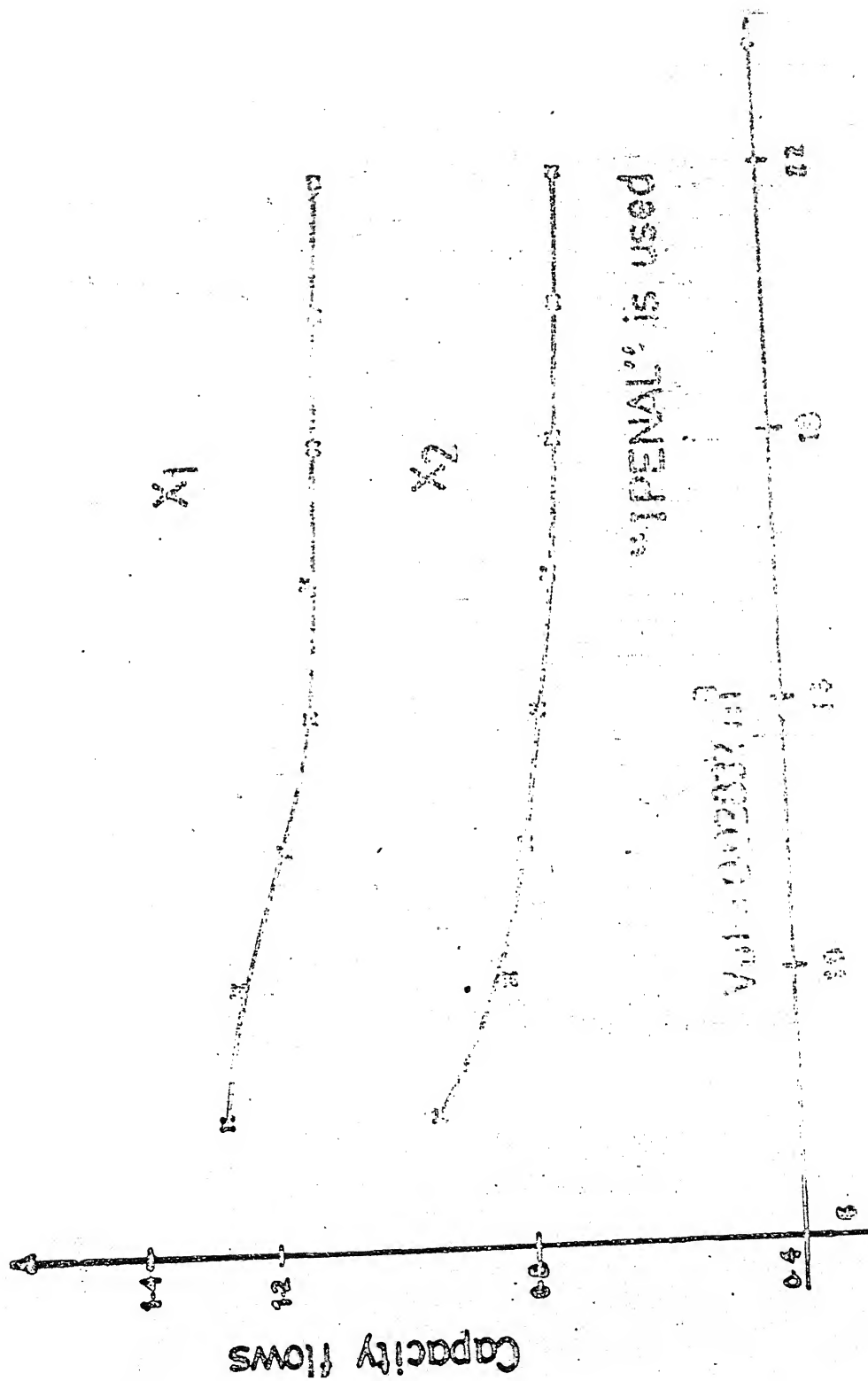


Fig 14. Optimum flows

It is obvious from Fig. 12 that the pumping power increases rapidly and the effectiveness decreases slowly as the volume decreases. The asterisk in Fig. 12 represents the pumping power and the volume corresponding to the unoptimized exchanger studied in reference [15]. For the same volume, the optimum design reduces the power ϕ from 31.02 HP to 9.03 HP. If however, the pumping power requirement is kept the same, say 31.02 HP, the volume of the optimized exchanger is only 0.1614 m^3 as compared to the 0.4785 m^3 , for the unoptimized case. Fig. 14, indicates the behaviour of the flow rates with volume of the heat exchanger. It is interesting to note that there is only a slight variation in the magnitude of the flow rates of both the fluids with volume.

5.2 A COMPARISON OF OPTIMIZATION METHODS

With the large number of optimization strategies available, the natural question arises - "which is the best strategy?" At present, a comparative study of optimization methods applied to heat exchanger problems, in particular, is not available. However, several studies have been performed to evaluate the effectiveness of various optimization techniques by applying them to specially prepared test problems given by Himmelblau [2]. Most heat exchanger optimization problems are constrained and, hence, the following observations are presented for

the constrained methods only.

- (i) The sequential penalty approach and a gradient based search are more efficient when a large number of design variables are involved. The sequential exterior and interior penalty function methods work well with the gradient based search procedures even when using finite difference gradients. Heat exchanger performance is not generally described by a differentiable function and, therefore, internal finite difference schemes are necessary. The exterior penalty function method does not require a starting point which satisfies all the constraints. This is a distinct advantage in starting a heat exchanger design optimization for an unfamiliar application.
- (ii) The method of feasible directions is rapid and effective in solving smaller problems, with perhaps upto three or four variables and having mostly linear constraints. This method, however, may have significant difficulty when a large number of design variables or nonlinear constraints are involved. Constraints on pressure drop heat transfer rate or weight, for example, are nonlinear. Geometric constraints are generally linear.

5.3 CONCLUSIONS

1. Well structured optimization programs-package has been developed and tested for it's reliability, flexibility and accuracy with standard functions, available in the open literature [1,2,3].

2. A direct transfer compact heat exchanger for use in gas turbines is designed for minimum pumping power subject to constraints on volume, dimensions, and heat transferred. The minimization problem, formulated in (13) in Chapter 4, is quite general and can be used for any heat transfer application. The ϵ -NTU approach is used for evaluating the heat exchanger performance. Polynomial expressions are found to approximate the experimental data for the fluid transport properties and the friction and colburn factors for the heat transfer surfaces.

3. The optimization study of the compact heat exchanger provides a design which has got much superior performance.

REFERENCES

1. FOX, R.L., Optimization Methods for Engineering Design, Addison-Wesley, 1971.
2. Himmelblau, D.M., Applied Non-Linear Programming, McGraw-Hill, 1972.
3. Rao, S.S., Optimization theory and applications, Wiley Eastern Ltd., 1979.
4. Smith, E.C., "Air-Cooled Heat Exchangers", Chemical Engineering, Nov. 17, 1958, pp 145-150.
5. Bergles, A.E., Blumenkrantz, A.R., and Taborek, J., Proceedings of the Fifth International Heat Transfer Conferences, Vol. II 1976, pp 239-263.
6. Fax, D.H., and Mills, R.R., Jr., "Generalized Optimal Heat Exchanger Design", ASME Trans, Vol 79, 1957, pp 653-661.
7. Briggs, D.E. and Evans, L.B., 49th AIChE National meeting, New Orleans, La, 1963.
8. Wilde, D.J., Industrial and Engineering Chemistry V.57, No. 3, 1965, pp 13-31.
9. Palen, J.W., Cham, T.P., and Taborek, J. "Optimization of Shell-and-Tube Heat Exchangers by Case Study Method", Chemical Engineering Progress Symposium Series, Vol.70, No. 138, 1974, pp 205-214.

10. Johnson, C.M., Vanderplaats, G.N., and Marto, P.J.,
"Marine Condenser Design Using Numerical Optimization",
Journal of Mechanical Design, Vol. 102, 1980, pp 469-475.
11. Fontein, H.J., and Wassink, J.G., "The Economically
Optimal Design of Heat Exchangers", Engineering
and Process Economics, Vol. 3, 1978, pp 141-149.
12. Shah, R.K., Afimiwala, K.A., Mayne, R.W., "Heat
Exchanger Optimization", Proceedings of the Sixth
International Heat Transfer Conference, Vol. 4, 1978,
pp 185-191.
13. Shah, R.K., "Compact Heat Exchanger Design Procedures",
Heat Exchangers - Thermohydraulic Fundamentals and
Design ed. by S.Kakac, A.E. Bergles, and F. Mayinger,
Hemisphere Publishing Corp., New York (1981) pp 495-
535.
14. Hedderich, C.P., Kelleher, M.D., Vanderplaats, G.N.,
"Design and Optimization of Air Cooled Heat Exchangers",
Journal of Heat Transfer Vol. 104, Nov. 1982, pp 633-
690.
15. Kays, W.M., and London, A.L., Compact Heat Exchangers,
2nd ed., McGraw-Hill, 1964.
16. Shah, R.K., "Classification of Heat Exchangers",
Heat Exchangers - Thermohydraulic Fundamentals and
Design ed. by S. Kakac, A.L. Bergles, and F. Mayinger,
Hemisphere Publishing Corp, New York (1981) pp 9-46.

17. Shah, R.K., " Compact Heat Exchanger Surface Selection Methods", Proceedings of the Sixth Intl Heat Transfer Conference, Toronto, Aug 1978, Vol. 4, pp 193-199.
18. Cox, B., and Jallouk, P.A., " Methods of Evaluating the Performances of Compact Heat Transfer Surfaces", ASME, Journal of Heat Transfer, Nov. 1973.
19. R.C. Reid and T.K. Sherwood, " The Properties of Gases and Liquids, Their Estimation and Correlation", 2nd edn. McGraw-Hill Book Coy, New York (1966).
20. Chapman, A.J., Heat Transfer, McMillan, New York. (1966)
21. Mason, J.L., " Heat Transfer in Cross-Flow", Proc. Appl. Mech. 2nd U.S. Congress, 1954, p 801.

APPENDIX - A : USER'S MANUAL

A.1 LIST OF SUBROUTINES

SUBROUTINES

PURPOSE

UNCONSTRAINED OPTIMIZATION :=

UNIV	Univariate search
STEEP	Steepest descent search
CONDIR	Conjugate directions search
	[Powell's Method]
CONGRA	Conjugate gradient search
	[Fletcher - Reeves Method]
DFPM	Variable Metric Method
	[Davidon - Fletcher - Powell Method]
QUAD	1 - D minimization by Quadratic
	interpolation scheme
GOLD	1 - D minimization by Golden section
	method

CONSTRAINED OPTIMIZATION :=

IPENAL	Interior penalty functions method
EXPEN	Exterior penalty functions method
FEAS	Method of feasible directions
START	Gets a feasible starting point

LINEAR PROGRAMMING :=

LINEAR	Linear programming by simplex method
--------	--------------------------------------

INTERNAL SUBROUTINES :=

CONSTR	Determines constraints violated
GRAD , GRAD1	Package routines for determining gradient by
	forward differences scheme
MODIFY	Modifies the objective combining the
	constraints
CONVRG	Checks for convergence by perturbation scheme

USER WRITTEN SUBROUTINES :=

FUNCT	User written function subroutine
GRADU	User written gradient subroutine

COMMON STATEMENTS

Some common statements have to be included in user written programmes

1. COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT
Must be included in the mainline.
2. COMMON/PENFUN/R,UF
Insert in mainline program and function subroutine
FUNCT when penalty functions are used
3. COMMON/CONVEC/G
DIMENSION G(25)
These statements must be included in function s
subroutine FUNCT if constraints are involved. The dimension
of G is always set to 25 in order to overcome the problem
of transferring the value of G from FUNCT to internal
subroutines.
4. COMMON/COMAT/DG
DIMENSION DG(-,-)
Placed in mainline program and used with gradient
subroutine GRADU if constraint gradients are needed as in
subroutines START,FEAS
Matrix DG must be properly dimensioned using a
appropriate integers. It's a matrix of order (NC,NN) , the
elements of which represent the components of the
constraints gradient.

A.3

EXAMPLES OF STANDARD TEST FUNCTIONS

C

TEST FUNCTION IS TAKEN FROM FOX [Ref 1] Page 89

COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT
COMMON/NEW/FF,ACC,IGRAD

DIMENSION X(2)

OPEN(UNIT=1,FILE='NN.')

NN=2

NC=0

NI=0

X(1)=0.

X(2)=0.

WRITE(1,5)(X(I),I=1,NN)

FORMAT('X(I) := ',9(1PE9.2))

MINIM=2

TMAX=.5

TI=.1

ACC=1.E-05

NPERT=1

LIM=15

METHOD=1

FF=.01

IGRAD=1

TO USE GRADU , IGRAD = 2

IDUT=4

KFEAS=1

CALL UNIV(MINIM,X,TT,NN,F)

WRITE(1,10)(X(I),I=1,2),F

FORMAT(/' [X] := ',2(1PE14.6)//' [F] := ',1PE14.6)

CLOSE(UNIT=1)

STOP

END

SUBROUTINE FUNCT(X,F)

DIMENSION X(2)

F=10.*X(1)^4-20.*X(2)*X(1)^2+10.*X(2)^2+X(1)^2-2.*X(1)+5.

RETURN

END

SUBROUTINE GRADJ(X,S,SUM)

DIMENSION X(2),S(2)

S(1)=-40.*X(1)^3+40*X(2)*X(1)-2.*X(1)+2.

S(2)=20.*X(1)^2-20.*X(2)

SUM=S(1)^2+S(2)^2

IF(SUM)1,2,1

SUM=1.

RETURN

END

RETURN

END

SOURCE := FOX [Ref : 1] PROBLEM F5 PP 89

NO OF VARIABLES := TWO

CONSTRAINTS := NONE

TEST FUNCTION :=

MINIMIZE

$$F(X) = 10.*X(1)^4 - 20.*X(2)*X(1)^2 + 10.*X(2)^2 + X(1)^2 - 2.*X(1) + 5.0$$

NOTE : THIS FUNCTION BELONGS TO ROSENBROCK'S BANANA VALLEY FUNCTION FAMILY

STARTING POINT :=

[X] := [0.0 , 0.0]

RESULT GIVEN :=

[X] := [1. , 1.]
F := 4.0

RESULTS OBTAINED:=

BY UNIV [X] := [9.9999 E-01 , 1.0000 E+00]
F := 4.0000 E+00

BY STEEP [X] := [9.9890 E-01 , 9.7736 E-01]
F := 4.0001 E+00

BY CONDIR [X] := [9.9990 E-01 , 9.9981 E-01]
F := 4.0000 E+00

BY CONGRA [X] := [8.1308 E-01 , 6.5341 E-01]
F := 4.0355 E+00

BY DFPM [X] := [9.4562 E-01 , 8.9172 E-01]
F := 4.0030 E+00

SOURCE := D.M.HIMMELBLAU [Ref : 2] PROBLEM NO 26 PP 427
 NO OF VARIABLES := FOUR
 CONSTRAINTS := NONE
 TEST FUNCTION :=
 MINIMIZE:

$$F(X) = (X(1)+10.*X(2))^2 + 5.*(X(3)-X(4))^2 + (X(2)-2.*X(3))^4 + 10.*(X(1)-X(4))^4$$

STARTING POINT :=

[X] := [3.0 , -1.0 , 0.0 , 1.0]

RESULT GIVEN :=

[X] := [0.0 , 0.0 , 0.0 , 0.0]
 F := 0.0

RESULTS OBTAINED:=

BY STEEP [X] := [1.544E-01, -1.617E-02, 7.949E-02, 8.235E-02]
 F := 1.444E-03

BY CONDIR [X] := [3.426E-02, -3.472E-03, 3.635E-02, 3.739E-02]
 F := 3.932E-05

BY CONGRA [X] := [1.048E-02, -1.038E-02, 5.859E-02, 5.943E-02]
 F := 3.114E-04

BY DFPM [X] := [1.585E-02, -1.661E-03, -1.457E-02, -1.418E-02]
 F := 1.056E-05

C

THIS PROBLEM IS TAKEN FROM S.M.HIMMELBLAU [Ref :2] NO:24 PP426

```

COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT
COMMON/PENFUN/R,UF
COMMON/NEW/FF,ACC,IGRAD
COMMON/CMAT/DG
DIMENSION X(2),DG(2,2)
OPEN(UNIT=1,FILE='NN.')

```

NN=2

NC=2

NI=2

X(1)=2.0

X(2)=2.0

WRITE(1,5)(X(I),I=1,NN)

5 FORMAT('X(I) := ',9(1PE9.2))

MINIM=2

TMAX=.5

TT=.1

ACC=1.E-05

NPERT=1

LIM=15

METHOD=5

FF=.01

IGRAD=1

C TO USE GRADU , IGRADU = 2

IDUT=6

KFEAS=2

FC=.1

R=1.

CALL IPENAL(MINIM,X,TT,NN,FC,F)

10 WRITE(1,10)(X(I),I=1,NN),F

FORMAT('///[X] := ',3(1PE14.6))//'[F] := ',1PE14.6)

CLOSE(UNIT=1)

STOP

END

SUBROUTINE FUNCT(X,F)

COMMON/PENFUN/R,UF

COMMON/CONVEC/G

DIMENSION X(3),G(25)

F=(X(1)-2.)² + (X(2)-1.)²G(1)=X(1)² - X(2)

G(2)=X(1) + X(2) - 2.

UF=F

RETURN

END

SUBROUTINE GRADU(X,S,SUM)

COMMON/CMAT/DG

DIMENSION X(2),S(2),DG(2,2)

S(1)=-2.*(X(1)-2.)

S(2)=-2.*(X(2)-1.)

SUM=S(1)²+S(2)²

IF(SUM)1,2,1

SUM=1.

2 DG(1,1)=1.*X(1)

1 DG(1,2)=-1.

DG(2,1)=1.

DG(2,2)=1.

RETURN

END

SOURCE := D.M.HIMMELBLAU [Ref : 2] PROBLEM NO 24 PP 426

NO OF VARIABLES := TWO

CONSTRAINTS :=

LINEAR INEQUALITY : ONE

NON LINEAR INEQUALITY : ONE

TEST FUNCTION :=

MINIMIZE

$$\begin{aligned} F(X) &= (X(1)-2.)^2 + (X(2)-1.)^2 \\ G(1) &= X(1)^2 - X(2) \leq 0.0 \\ G(2) &= X(1) + X(2) - 2. \leq 0.0 \end{aligned}$$

STARTING POINT :=

[X] := [2.0 , 2.0]

RESULT GIVEN :=

[X] := [1. , 1.]
F := 1.0

RESULTS OBTAINED:=

DFPM IS USED IN ALL THE CASES

BY IPENAL

[X] := [1.0001 E+00 , 1.0005 E+00]
F := 9.9968 E-01

BY EXPEN

[X] := [1.0001 E+00 , 1.0005 E+00]
F := 9.9968 E-01

BY FEAS

[X] := [9.9795 E-01 , 9.9795 E-01]
F := 1.0041 E+00

SOURCE := S.S.RAD [Ref : 3] PROBLEM 7.7 PP 391
NO OF VARIABLES := THREE
CONSTRAINTS :=

LINEAR INEQUALITY : ONE
NON LINEAR INEQUALITY : TWO
BOUNDS ON INDEPENDENT : THREE
VARIABLES

TEST FUNCTION :=

MINIMIZE.

$F(X) = X(1)^3 - 6.*X(1)^2 + 11.*X(1) + X(3)$

NOTE : X(2) IS NOT INCLUDED IN F(X) DEFINITION

$G(1) = X(1)^2 + X(2)^2 - X(3)^2 \leq 0.0$
 $G(2) = 4.0 - X(1)^2 - X(2)^2 - X(3)^2 \leq 0.0$
 $G(3) = X(3) - 5.0 \leq 0.0$
 $G(4) = -X(1) \leq 0.0$
 $G(5) = -X(2) \leq 0.0$
 $G(6) = -X(3) \leq 0.0$

STARTING POINT :=

[X] := [0.1 , 0.1 , 3.0]

RESULT GIVEN :=

[X] := [3.248E-08 , 1.41421 , 1.41421]
F := 1.41422E+00

RESULTS OBTAINED:=

DFPM IS USED IN ALL THE CASES

BY IPENAL [X] := [1.00E-05 , 1.41465E+00 , 1.41467E+00]
F := 1.447821E+00

BY EXPEN [X] := [5.5269E-04 , 1.40193E-02 , 1.99367E+00]
F := 1.99368E+00

BY FEAS [X] := [0.0000E+00 , 1.41422E+00 , 1.41422E+00]
F := 1.41422E+00

SOURCE := D.M.HIMMELBLAU [Ref : 2] PROBLEM NO 11 PP 406

NO OF VARIABLES := FIVE

CONSTRAINTS :=

LINEAR INEQUALITY : 6

BOUNDS ON INDEPENDENT

VARIABLES : 10

TEST FUNCTION :=

MINIMIZE

$F(X) = 5.3578547 * X(3)^2 + .8356891 * X(1) * X(5) + 37.293239 * X(2) - 40792.141$

NOTE : X(2) & X(4) AREN'T INCLUDED IN F(X) DEFINITION

CONSTRAINTS ARE :=

$0.0 \leq 85.334407 + 0.0056858 * X(2) * X(5) + 0.0006252 * X(1) * X(4) - 0.0022053 * X(3) * X(5) \leq 92.0$

$90.0 \leq 80.51249 + 0.0071317 * X(2) * X(5) + 0.0029955 * X(1) * X(2) - 0.0021813 * X(3)^2 \leq 110.0$

$20.0 \leq 9.300961 + 0.0047026 * X(3) * X(5) + 0.0012547 * X(1) * X(3) - 0.0019085 * X(3) * X(4) \leq 25.0$

$78.0 \leq X(1) \leq 102.0$

$33.0 \leq X(2) \leq 45.0$

$27.0 \leq X(3) \leq 45.0$

$27.0 \leq X(4) \leq 45.0$

$27.0 \leq X(5) \leq 45.0$

STARTING POINT :=

[X] := [78.62 , 33.44 , 31.07 , 44.18 , 35.22]

RESULT GIVEN :=

[X] := [78.003, 29.995, 45.000, 36.776]
F := - 30665.5

RESULTS OBTAINED:=

BY DFPM

[X] := [78.595, 33.317, 30.739, 44.162, 35.129]
F := - 30493.3

A.4 COMPACT HEAT EXCHANGER PROBLEM

```

COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IOUT,NPERT
COMMON/PENFUN/R,UF
COMMON/NEW/FF,ACC,IGRAD
COMMON/CALL/NCALL,VMAX,BIG,VOL
COMMON/PARA/YY
COMMON/EXCHGR/EF
DIMENSION YY(5)
OPEN(UNIT=1,FILE='R.')
CALL RTIME(MI)
NCALL=0
NN=5
NC=12
NI=12
YY(1)=2.5
YY(2)=1.
YY(3)=.5
YY(4)=.3
YY(5)=1.25
VMAX=14.0
BIG=5.
5 WRITE(1,5)(YY(I),I=1,5)
  FORMAT(' [ X ] := ',9(1PE9.2))
6 WRITE(1,6)BIG,VMAX
  FORMAT(' MAX FOLW RATE := ',F5.2,' V max := ',F5.2)
MINIM=2
TMAX=.5
TT=.1
ACC=1.E-05
NPERT=1
LIM=15
METHOD=5
FF=.01
IGRAD=1
IOUT=6
KFEAS=1
FC=10.
R=10.
CALL EXPEN(MINIM,YY,TT,NN,FC,F)
11 WRITE(1,11)EF,NCALL
  FORMAT('EFFECTIVENESS ',1PE14.6/'NO OF FUNCTION CALLS ',I5)
12 WRITE(1,12)VOL
  FORMAT('VOLUME := ',1PE14.6)
  CALL RTIME(MF)
  MT=MF-MI
10 WRITE(1,10)MT
  FORMAT('CPU TIME := ',I5,'m.sec'/)
  CLOSE(UNIT=1)
  TYPE 100
100 FORMAT(' WANT GRAPHIC OPTION ?(Y/N) '$)
  ACCEPT 101,ANS
101 IF(ANS.NE.'Y')GOTO 102
  CALL GRAPH
  YOU HAVE TO ASSIGN TTY 5 FOR GRAPHIC OPTION
102 STOP
  END

```

```

C      SUBROUTINE FUNCT(Y,Y,F)
C      -----
C      HEAT EXCHANGER PERFORMANCE ANALYSIS ROUTINE DEFINING THE
C      OBJECTIVE FUNCTION
COMMON/PENFUN/RR,UF
COMMON/CONVEC/G
COMMON/CALL/NCALL,VMAX,BIG,VOL
COMMON/EXCHGR/EF
DIMENSION YY(5),G(25)
REAL NUC,NTU,L
NCALL=NCALL+1
DO 5 I=1,5
IF(YY(I))100,100,5
5  CONTINUE
THI=260.
TCI=60.
PA=39.7
PW=14.7
C      QD=8.E+06
SURFACE CHAR
RH=.00288
ALFAH=270.
SIGMAH=.78
RC=.00306
ALFAC=42.1
SIGMAC=.129
C      FIN CHAR
FA=.845
DEL=.00033
FL=.01875
FK=100.
NON DIMEN
L=VMAX*(1./3.)
Z=QD/(THI-TCI)
C      SOLUTION
AFRH=YY(3)*YY(5)*L^2.
AFRC=YY(3)*YY(4)*L^2.
VOL=YY(3)*YY(4)*YY(5)*VMAX
E1=.75
CH=YY(1)*Z
CC=YY(2)*Z*10.
CMIN=AMIN1(CH,CC)
CMAX=AMAX1(CH,CC)
9  E=E1
Q=CMIN*E*(THI-TCI)
THO=THI-E*Q/CH
TCO=TCI+E*Q/CC
THAVG=(THI+THO)/2.
TCAVG=(TCI+TCO)/2.
CALL PROPC(TCAVG,RHOC,CONDC,CPC,UC)
CALL PROPH(THAVG,RHOF,CONDF,CPH,UH)
V1=53.345*(460.+THI)/(144.*PA)
V2=53.345*(460.+THO)/(144.*PA)
VM=53.345*(460.+THAVG)/(144.*PA)
GH=CH/(CPH*SIGMAH*AFRH)
GC=CC/(CPC*SIGMAC*AFRC)
REC=GC*4.*RC/UC
REH=GH*4.*RH/UH
PRC=UC*CPC/CONDC
PRH=UH*CPH/CONDF
C      DETERMINE COLBURN and FRICTION FACTOR FROM POLYNOMIAL FITS
Y=ALOG10(REH*.001)
FH=-.11998*Y^4.-.044628*Y^3+.26677*Y^2.-.45965*Y+.58734
COH=-.11873*Y^4+.035861*Y^3+.22707*Y^2.-.47894*Y-.0062232
FH=10.*(FH)*.01
COH=10.*(COH)*.01
HH=COH*GH*CPH/PRH^(2./3.)
Y=ALOG10(REC*.001)
IF(Y.LE.0.)GOTO 1
IF(Y.GE.1.)GOTO 2
FC=-1.2239*Y^4.+2.1643*Y^3.-.56969*Y^2.-.87268*Y+.38629
COC=-1.3431*Y^4.+1.6032*Y^3+.55494*Y^2.-1.2064*Y-.083788
GOTO 3
1  FC=.3802-.8745*Y
COC=-.09691-Y
GOTO 3

```

```

COC=10.*(COC)*.01
HC=COC*GC*CPC/PRC^(2./3.)
FM=SQRT(2.*HH/(FK*DEL))
Y=FM*FL
FINEFF=TANH(Y)/Y
EFF=1.-FA*(1.-FINEFF)
UCOND=EFF*HH*(ALFAC/ALFAH)*HC/(ALFAC/ALFAH*HC+EFF*HH)
NTU=ALFAH*VOL*UCOND/CMIN
R=CMIN/CMAX
CALL EFFECT(E1,NTU,R)
IF(ABS(E-E1).GT..01)GOTO 9
Y=GH/3600.
PH=Y*.2.*V1*((1.+SIGMAH^2.)*(V2/V1-1.)+FH*YY(4)*L*(VM/V1)/RH)
1 /64.4
Y=GC/3600.
PC=Y*.2.*FC*YY(5)*L/(RHOC*RC*64.4)
POWER=(PH*CH*VM/CPH+PC*CC/(RHOC*CPC))/(550.*3600.)
P=POWER
UF=F
EF=E1
G(1)=-YY(1)
G(2)=YY(1)/BIG-1.
G(3)=-YY(2)
G(4)=YY(2)/(10.*BIG)-1.
G(5)=-YY(3)
G(6)=-YY(4)
G(7)=-YY(5)
G(8)=L*YY(4)-1.
G(9)=YY(3)/(2.*YY(4))-1.
G(10)=YY(5)/(3.*YY(3))-1.
G(11)=VOL/VMAX-1.
G(12)=1.-Q/QD
RETURN
END

```



```

C      SUBROUTINE PROPC(T,R,C,CP,U)
POLYNOMIAL FITS FOR TRANSPORT PROPERTIES OF AIR
X=T/100.
CP=.0017822*X^4.-.017893*X^3+.067628*X^2-.091686*X+1.0348
R=10.*(-.0019308*X^4+.0218*X^3-.13*X^2+.080173*X+6.2283)
U=.045473*X^4.-.56414*X^3+.2.6107*X^2.-5.5424*X+5.1444
C=.1*(-.0033219*X^4+.043706*X^3.-.29151*X^2+.88891*X+3.0051)
RETURN
END

C      SUBROUTINE PROPH(T,R,C,CP,U)
POLYNOMIAL FITS FOR TRANSPORT PROPERTIES OF WATER
X=T/100.
U=.1*(-.00570913*X^4+.05755772*X^3.-.22800553*X^2 +
1      +.6754148*X+.40343288)
CP=.1*(.03306576*X^4.-.19582903*X^3+.30682039*X^2. +
1      +.20886326*X + 2.36525726)
C=.01*(.00734717*X^4.-.04445982*X^3.-.10930872*X^2. +
1      + 2.0579495*X +1.48826838)
RETURN
END

C      SUBROUTINE EFFECT(E,NTU,R)
EFFECTIVENESS FOR CROSS-FLOW EXCHANGER WITH BOTH FLUIDS UNMIXED
REAL NTU
SUM=(1.-EXP(-NTU))*(1.-EXP(-NTU*R))
SUMO=SUM
N=1
3      SUM1=1.
SUM2=1.
SM=1.
DO 1 J=1,N
SM=SM*J
SUM1=SUM1+(NTU)^J/SM
SUM2=SUM2+(NTU*R)^J/SM
1      CONTINUE
SUM=SUM+(1.-EXP(-NTU)*SUM1)*(1.-EXP(-NTU*R)*SUM2)
IF(ABS(SUM-SUMO).LT.1.E-05)GOTO 2
N=N+1
SUMO=SUM
GOTO 3
2      E=SUM/(NTU*R)
RETURN
END

```

SUBROUTINE GRAPH

 THIS ROUTINE USES GPGS ROUTINES FOR GRAPHIC PURPOSES

```

COMMON/PARA/YY
COMMON/TRANS/W1,W2,W3
COMMON/EXCGHR/EFF
COMMON/ACCEPT/CH,CC,X,Y,Z
DIMENSION V1(6),V2(6),V3(6),W1(6),W2(6),W3(6),YY(5)
DATA V1/0.,1.3,0.,1.0,1.,1./
DATA V2/0.,1.0,1.,1.0,1.,1./
DATA V3/1.,1.3,1.,1.1,1.,1./
DATA W1/-10.,1303.,-10.,1000.,-10.,1000./
DATA W2/-10.,900.,-10.,900.,-10.,1000./
DATA W3/901.,1204.,-10.,1000.,-10.,1000./
CH=YY(1)*40000.
CC=YY(2)*400000.
X=YY(3)
Y=YY(5)
Z=YY(4)
CALL NITDEV(5)
CALL CLRDEV(5)
CALL VPORT3(V1)
CALL WINDW3(W1)
TYPE 5
FORMAT(6('/))' WANT CAPTION ? (Y/N) 's)
ACCEPT 6,CAP
FORMAT(A1)
IF(CAP.NE.'Y')GOTO 7
CALL CLRDEV(5)
CALL TITLE
TYPE 40
FORMAT(5('/))'!!! RETURN FOR CONTINUING !! 's)
ACCEPT 15,TTY
FORMAT(A4)
IF(TTY.EQ.'STOP')GOTO 100
CALL CLRDEV(5)
CALL MARGIN
CALL VPORT3(V3)
CALL WINDW3(W3)
CALL LINE3(W3(1),W3(4),0.,0)
CALL LINER3(0.,-W3(4),0.,1)
WC1=W3(1)+4.0
WC2=W3(4)-100.
CALL SOFCTL(0)
CALL CHAR('LIKE TO CHANGE*')
CALL LINE3(WC1,WC2-50.,0.,0)
CALL CHAR('WINDW SIZE(Y/N)*')
ACCEPT 31,A1
FORMAT(A1)
IF(A1.NE.'Y')GOTO 32
CALL LINE3(WC1,WC2-100.,0.,0)
CALL CHAR('GIVE ME W2(1:6)*')
CALL LINE3(WC1,WC2-150.,0.,0)
ACCEPT *,(W2(I),I=1,6)
CALL CLRDEV(5)
CALL VPORT3(V1)
CALL WINDW3(W1)
CALL MARGIN
CALL WINDW3(W1)
CALL VPORT3(V2)
CALL ROTAD(30.,2)
CALL ROTAD(-30.,1)
CALL OBJECT
CALL IDEN
CALL DIMEN
CALL VPORT3(V3)
CALL WINDW3(W3)
WF=W3(4)*.6
CALL LINE3(W3(1),WF,0.,0)
CALL CHAR('WANT REPEAT(Y/N)*')
ACCEPT 12,ANS
FORMAT(A1)
IF(ANS.NE.'Y')GOTO 100
CALL VPORT3(V1)
CALL WINDW3(W1)
GOTO 1
CALL CLRDEV(5)

```

```

SUBROUTINE OBJECT
COMMON/ACCEPT/CH,CC,X,Y,Z
CALL LINE3(10.,0.,900.,0.,0.,0.)
CALL LINER3(X,0.,0.,0.,0.,1)
CALL LINER3(0.,0.,0.,0.,Z,1)
CALL LINER3(-X,0.,0.,0.,0.,1)
CALL LINER3(0.,0.,0.,0.,-Z,1)
CALL LINER3(0.,0.,-Y,0.,0.,1)
CALL LINER3(X,0.,0.,0.,0.,1)
CALL LINER3(0.,0.,0.,0.,Z,1)
CALL LINER3(0.,0.,Y,0.,0.,1)
CALL LINER3(0.,0.,0.,0.,-Z,1)
CALL LINER3(0.,0.,-Y,0.,0.,1)
FIN=.025
T=Y*FIN
T GIVES FIN THICKNESS
N=IFIX(Y/(2.*T))
CALL LINE3(10.,900.,0.,0.)
DO 10 I=1,N
CALL LINER3(0.,0.,-T,0.,0.,0)
CALL LINER3(X,0.,0.,0.,0.,1)
CALL LINER3(0.,0.,0.,0.,Z,1)
CALL LINER3(0.,0.,-T,0.,0.,0)
CALL LINER3(0.,0.,0.,0.,-Z,1)
CALL LINER3(-X,0.,0.,0.,0.,1)
CONTINUE
TT=902.-Y
CALL LINE3(10.,0.,TT,0.,0.,0)
CALL LINER3(X,0.,0.,0.,0.,1)
CALL LINER3(0.,0.,0.,0.,Z,1)
RETURN
END

```

```

SUBROUTINE MARGIN
COMMON/TRANS/W1,W2,W3
DIMENSION W1(6),W2(6),W3(6)
CALL LINE3(W1(1),W1(4),0.,0)
CALL LINER3(W1(2),0.,0.,0.,1)
CALL LINER3(0.,-W1(4),0.,1)
CALL LINER3(-W1(2),0.,0.,1)
CALL LINER3(0.,W1(4),0.,1)
STOP
END

```

```

SUBROUTINE TITLE
COMMON/TRANS/W1,W2,W3
DIMENSION W1(6),W2(6),W3(6)
DIMENSION C1(9),C2(7),C3(7)
CALL SOFCTL(1)
CALL CSIZEV(15)
DATA C1/'OPTIMIZATION METHODS FOR ENGINEERING DESIGN*.'/
DATA C2/'For COMPACT HEAT - EXCHANGERS
DATA C3/'Staff Advisor : Prof H.C.Agrawal*.'/
CALL LINE3(W1(1),W1(4),0.,0)
CALL LINER3(W1(2),0.,0.,0.,1)
CALL LINER3(0.,W1(4),0.,1)
CALL LINER3(-W1(2),0.,0.,1)
CALL LINER3(0.,W1(4),0.,1)
WI=200.
WW=100.
WY=W1(4)-200.
CALL LINE3(WW,WY,0.,0)
CALL CHAR(C1,45)
CALL LINE3(WW,WY-200.,0.,0)
CALL CHAR(C2,31)
CALL LINE3(WW,WY-400.,0.,0)
CALL LINE3(WW,WY-400.,0.,0)
CALL CHAR(C3,34)
RETURN
END

```

```

SUBROUTINE DIMEN
COMMON/TRANS/W1,W2,W3
COMMON/ACCEPT/CH,CC,X,Y,Z
COMMON/EXCGHR/EFF
DIMENSION W1(6),W2(6),W3(6)
CALL SOFCTL(1)
CALL LINE3(W2(1)+10.,W2(4)-50.,0.,0)
CALL CSIZEH(30)
CALL CSIZEV(18)
CALL CSHEA(.2)
CALL CHAR('OPTIMIZED DIMENSIONS*.*')
CALL SOFCTL(0)
D1=W2(2)*.5
D2=W2(4)-200.
CALL LINE3(D1,D2,0.,0)
CALL CHAR('WIDTH :=*.*')
CALL CHARF(X,7,2)
CALL LINE3(D1,D2-50.,0.,0)
CALL CHAR('BREADTH :=*.*')
CALL CHARF(Y,7,2)
CALL LINE3(D1,D2-100.,0.,0)
CALL CHAR('HEIGHT :=*.*')
CALL CHARF(Z,8,2)
CALL LINE3(D1,D2-150.,0.,0)
CALL CHAR('AIRFLOW :=*.*')
CALL CHARF(CH,9,1)
CALL LINE3(D1,D2-200.,0.,0)
CALL CHAR('WATER FLOW :=*.*')
CALL CHARF(CC,9,1)
CALL LINE3(D1,D2-250.,0.,0)
CALL CHAR('EFFECTIVENESS :=*.*')
CALL CHARF(EFF,6,4)
RETURN
END

```

RESULT OF TRADE-OFF STUDIES

VOLUME	METHOD	F	X1	X2	E
8	IPENAL	21.684	1.2745	0.9431	0.7842
	EXPEN	21.548	1.2870	0.8509	0.7763
	FEAS	21.700	1.2170	0.9540	0.7842
10	IPENAL	15.173	1.2106	0.8171	0.8251
	EXPEN	15.162	1.2116	0.8138	0.8243
	FEAS	15.400	1.1940	0.9520	0.8356
12	IPENAL	11.723	1.1648	0.7949	0.8571
	EXPEN	11.716	1.1650	0.7943	0.8570
	FEAS	12.030	1.1440	0.9550	0.8710
14	IPENAL	9.592	1.1341	0.6777	0.8800
	EXPEN	9.606	1.1286	0.8039	0.8840
	FEAS	9.770	1.1170	0.8990	0.8920
16	IPENAL	8.155	1.1123	0.7423	0.8971
	EXPEN	8.126	1.1111	0.7389	0.8969
	FEAS	8.550	1.0880	0.9610	0.9153
18	IPENAL	7.125	1.0947	0.7284	0.9114
	EXPEN	7.103	1.0946	0.7190	0.9503
	FEAS	6.190	1.0480	0.9670	0.9503
20	IPENAL	6.351	1.0812	0.7139	0.9227
	EXPEN	6.334	1.0819	0.6995	0.9212
	FEAS	6.781	1.0580	0.9650	0.9412
22	IPENAL	5.750	1.0704	0.7011	0.9319
	EXPEN	5.736	1.0718	0.6814	0.9299
	FEAS	5.723	1.0480	0.9667	0.9502

APPENDIX - B : PROGRAMS PACKAGE LISTINGS

```

=====
SUBROUTINE RANGE(NN,NQUIT,NREP,IT,TT1,X,S,AF,AL,XH,XL,Z)
=====
THIS SUBROUTINE DETERMINES THE RANGE WHICH CONTAINS THE MINIMUM
NQUIT=1 RANGE FOR MINIMUM LOCATED
NQUIT=2 CONSTRAINT IN EFFECT
NQUIT=3 NO IMPROVEMENT POSSIBLE IN THE OBJECTIVE FUNCTION
NREP=1 NO. OF ITERATIONS INADEQUATE TO ESTIMATE THE RANGE
NREP=2 RANGE ESTIMATED
NREP=3 NO IMPROVEMENT IN OBJECTIVE FUNCTION POSSIBLE
IT,IT1 = INITIAL STEP SIZE
TMAX = MAX STEP SIZE ALLOWED
AL,AF = COORD. AND FUNCTION VALUES FOR QUAD. INTERPOLATION
Z=INITIAL AND FINAL VALUES ALONG A SEARCH DIRECTION
KFEAS=1 UNCONSTRAINED OR EXTERIOR PENALTY FUNCTION PROBLEM
KFEAS=2 UNCONSTRAINED PROBLEM USING INTERIOR PENALTY FUNCTION
KFEAS=3 CONSTRAINED PROBLEM USING FEASIBLE DIRECTION METHOD
METHOD = CHOICE OF MULTIVARIATE SEARCH TECHNIQUE
METHOD=1 UNIVARIATE SEARCH
METHOD=2 STEEPEST DESCENT
METHOD=3 CONJUGATE DIRECTIONS [ POWELL'S METHOD ]
METHOD=4 CONJUGATE GRADIENT [ FLETCHER REEVES METHOD ]
METHOD=5 VARIABLE METRIC [ DAVIDON FLETCHER POWELL METHOD ]
IDOUT = TYPE OF OUTPUT DESIRED
NOTE := HIGHER ORDERS HAVE HIGHLY SUPPRESSED BUT
IMPORTANT RESULTS
IDOUT=1 OUTPUT AT ALL STAGES
IDOUT=2 OUTPUT AFTER EACH 10 MINIMIZATIONS
IDOUT=3 FINAL OUTPUT OF UNCONSTAINED MINIMIZATION
IDOUT=4 OUTPUT AT ALL STAGES CONSTRAINED MINIMIZATION
IDOUT=5 FINAL OUTPUT OF EACH CONSTRAINED MINIMIZATION
IDOUT=6 ONLY FINAL OUTPUT OF CONSTRAINED MINIMIZATION
=====

```

```

COMMON/MIN/AAL
COMMON/CONVEC/G
COMMON/INCDVC/GO
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDOUT,NPERT
COMMON/PENFUN/R,UF
COMMON/NEW/FP,ACC,IGRAD
COMMON/ALARM/KAL,JAW
DIMENSION X(NN),S(50),AF(3),AL(3),Z(2)
DIMENSION YX(25),G(25),GO(25)
DIMENSION DG(25,25)
KNA=1
AAL=IT
IF(IDOUT.GT.2)GOTO 555
WRITE(1,305)KAL,JAW
WRITE(1,300)
WRITE(1,3)(X(J),J=1,NN)
WRITE(1,303)Z(1)
WRITE(1,304)(S(J),J=1,NN)
305 FORMAT(/35('-'))// 'ITERATION NO ',I2,' CYCLE NO ',I2/35('-')
300 FORMAT('INITIAL VALUES OF X(J):')
303 FORMAT('INITIAL VALUE OF THE OBJECTIVE FUNCTION'/1PE14.6)
304 FORMAT('INITIAL VALUES OF S(J)'/9(1PE14.6))
IF(NC)GOTO 555,555,577
WRITE(1,572)
572 FORMAT('INITIAL VALUES OF THE CONSTRAINTS')
WRITE(1,3)(GO(J),J=1,NC)
555 H=Z(1)
IF(NC)GOTO 570,570,571
571 DO 596 J=1,NC
596 G(J)=GO(J)
IF(KFEAS=2)GOTO 590,591,591
CALL CONSTR(NPENAL)
IF(NPENAL=1)GOTO 590,590,592
591 WRITE(1,593)KAL,JAW
592 FORMAT('* X(J) INFEASIBLE IN',I3,'th ITERATION OF '
593 1, I2, ' CYCLE *')
WRITE(1,3)(GO(J),J=1,NC)
595 CALL START(X,NN,NC,TT,DG,IDOUT)
590 CONTINUE
IF(KFEAS=2)GOTO 581,581,570
IF(IDOUT=1)GOTO 582,582,569
581 WRITE(1,583)
582 FORMAT(/'NPENAL',2X,'KNN',6X,'STEP SIZE',8X,'UNAUGMENTED',
583 18X,'FUNCTION',16X,'DESIGN VECTOR')
GOTO 569
570 IF(IDOUT=1)GOTO 568,568,569
568 WRITE(1,562)

```

```

KPE=1
KZ=1
AL(1)=0.
AF(1)=Z(1)
KZ=KZ+1
KN=1
KK=1
KNN=1
KZ1=1
IF(KZ1-LIM) 23,23,24
IT=IT+1
CHECK IF NO. OF ITERATIONS XDS THE MAX. SPECIFIED.
KPENAL=1. MAX NO. OF ITERATIONS INADEQUATE TO LOCATE THE MIN.
KPENAL=2. CONSTRAINT VIOLATED AFTER INITIAL DECREASE IN FUNCN.
KPENAL=3. CONSTRAINT VIOLATED RIGHT FROM START.
GOTO(102,22,122)KPENAL
DO 28 J=1,NN
X(J)=YX(J)
IF(NCD) 514,514,551
DO 103 J=1,NC
GO(J)=G(J)
GOTO 514
DO 26 J=1,NN
X(J)=X(J)+AALJ*S(J)
IF(NCD) 514,514,573
DO 574 J=1,NC
G(J)=GO(J)
GOTO 514
DO 13 J=1,NN
YX(J)=X(J)+AAL*S(J)
CALL FUNCT(YX,F)
CALL MODIFY(F)
CALL CONSTR(NPENAL)
IF(IOUT-1) 556,556,557
IF(NCD) 584,584,585
IF(KFEAS-2) 587,587,584
WRITE(1,586)NPENAL,KNN,AAL,UF,F,(YX(J),J=1,NN)
FORRAT(2)(14,2X),3(1PE14.6,2X),5(5(1PE14.6),/,60X))
GOTO 557
WRITE(1,7)NPENAL,KNN,AAL,F,(YX(J),J=1,NN)
FORRAT(X,14,2X,14,2X,1PE14.6,2X,1PE14.6,2X,5(6(1PE14.6),/44X))
IF(NPENAL-1) 60,60,61
IF(KNN-1) 62,62,63
KPENAL=3
IF(KFEAS-2) 64,64,65
IT=.5*IT
AAL=IT
KZ1=KZ1+1
GOTO 16
EXTRAPOLATION SCHEME TO LOCATE THE CONSTRAINT.
IF(F-H) 66,66,64
DO 67 K=1,NI
IF(G(K)) 67,68,68
CONTINUE
D=G(K)
MC=K
DO 69 K=1,NC
IF(G(K)) 69,69,70
IF(G(K)-D) 69,69,71
D=G(K)
MC=K
CONTINUE
AAL=(GO(MC)*AAL-G(MC)*H1)/(GO(MC)-G(MC))
DO 80 J=1,NN
YX(J)=X(J)+AAL*S(J)
CALL FUNCT(YX,F)
CALL MODIFY(F)
CALL CONSTR(NPENAL)
IF(IOUT-1) 558,558,559
IF(KFEAS-2) 588,588,589
WRITE(1,586)NPENAL,KNN,AAL,UF,F,(YX(J),J=1,NN)
GOTO 559
WRITE(1,7)NPENAL,KNN,AAL,F,(YX(J),J=1,NN)
IF(G(MC)-.1E-10) 72,73,74
IF(ABS(G(MC))-ACC) 73,73,75
IF(NPENAL-1) 76,76,74
IT=.5*(AAL-AALO)
KPE=2
GOTO 60
IF(NPENAL-1) 78,78,74
IF(KZ1-LIM) 77,77,24

```



```

104 DO 104 J=1,NC
GO(J)=G(J)
Z(1)=F
63 GOTO 122
KPENAL=2
IF(KFEAS-2)82,82,83
C STEP SIZE CHANGE. CONSTRAINT VIOLATED AFTER INITIAL DECREASE
C OF OBJECTIVE FUNCTION.
82 FT=AAL-AALO
AAL=AAL-TT
TT=.5*TT
AAL=AAL+TT
KPE=2
KZ1=KZ1+1
83 GOTO 16
60 IF(F-H)66,66,82
AALO=AAL
KPENAL=1
Z(2)=F
14 IF(Z(2)-H)14,14,15
H=Z(2)
H1=AAL
IF(NCD)552,552,553
553 DO 90 J=1,NC
90 GO(J)=G(J)
C DETERMINATION AND UPDATING OF AF AND AL. COMPARISON WITH MAX.
C STEP SIZE SPECIFIED.
552 IF(KZ-3)530,530,531
530 AF(KZ)=Z(2)
AL(KZ)=AAL
KZ=KZ+1
531 GOTO 532
AF(1)=AF(2)
AF(2)=AF(3)
AF(3)=Z(2)
AL(1)=AL(2)
AL(2)=AL(3)
AL(3)=AAL
532 CONTINUE
IF(KPE-1)49,49,48
49 IF(KNN-3)536,536,537
536 TT=AAL*(KNN+1)
AAL=TT
537 GOTO 50
50 AAL=2.*AAL
ST=AAL-AALO
IF(ST-IMAX)51,51,52
52 ST=IMAX
AAL=AALO+ST
GOTO 51
48 AAL=AAL+TT
51 CONTINUE
KNN=KNN+1
KK=KK+1
KZ1=KZ1+1
GOTO 16
C REVERSAL OF SEARCH DIRECTION. DECREASE OF STEP SIZE.
15 IF(KK-1)18,18,17
18 IF(KN-1)46,46,47
46 GOTO(20,47,20,47,47),METHOD
20 DO 19 J=1,NN
19 S(J)=-S(J)
KN=KN+1
GOTO 16
17 IF(KZ-3)533,533,534
533 AF(KZ)=F
AL(KZ)=AAL
534 GOTO 535
AF(1)=AF(2)
AF(2)=AF(3)
AF(3)=F
AL(1)=AL(2)
AL(2)=AL(3)
AL(3)=AAL
535 XL=AL(1)
XH=AL(3)
NREP=2
NQUIT=1
IF(TT-TT1)500,500,502
502 TT=TT1
GOTO 500
47 IF(KNA-5)151,151,152

```

```

KNA=KNA+1
GOTO 16
NQUIT=2
122 IT=IT1
IF(NCD)153,153,575
DO 576 J=1,NC
575 G(J)=GD(J)
576 GOTO 153
NREP=1
514 NQUIT=1
IF(IT-IT1)500,500,503
503 IT=IT1
GOTO 500
NQUIT=3
152 IF(NCD)153,153,579
DO 580 J=1,NC
579 G(J)=GD(J)
580 Z(2)=Z(1)
153 NREP=3
F=Z(1)
FORMAT(9(1PE14.6))
3 IF(LOUT.GT.2)GOTO 561
500 WRITE(1,501)
560 FORMAT('VALUES OF S(J) AT EXIT FROM RANGE ARE')
501 WRITE(1,3)(S(J),J=1,NN)
WRITE(1,85)NREP,NQUIT
85 FORMAT(2X,'NREP',2X,'NQUIT'/3X,I2,5X,I2)
WRITE(1,92)(G(J),J=1,NC)
92 FORMAT('G FROM RANGE AT EXIT'/9(1PE14.6))
561 RETURN
END

```

```

=====
SUBROUTINE QUAD(X,S,NN,AL,AF,F)
=====
COMPUTATION OF MINIMUM USING QUADRATIC INTERPOLATION
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
ACC -- ACCURACY DESIRED
S -- SEARCH DIRECTION VECTOR
F -- FUNCTION VALUE AT THE MINIMUM
LIM -- MAX NO. OF ITERATIONS
AL -- COORDINATES OF THREE POINTS
AF -- FUNCTION VALUES AT THESE THREE POINTS
=====
COMMON/PENFUN/R,UF
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT
COMMON/MIN/AAL
COMMON/NEW/FF,ACC,IGRAD
COMMON/ALARM/KAL,JAW
DIMENSION X(NN),S(50),AL(3),AF(3)
DIMENSION ZZ(25)
IF(IDUT-1)44,44,45
44 WRITE(1,45)
46 FORMAT('RANGE RESULTS FOR QUADRATIC APPROXIMATION')
WRITE(1,3)(AF(J),J=1,3)
WRITE(1,3)(AL(J),J=1,3)
3 FORMAT(3(1PE14.5))
78 IF(NC)78,78,79
43 WRITE(1,43)
FORMAT('RESULTS FROM QUADRATIC INTERPOLATION'/2X,'NO',4X,
1'STEPSIZE',9X,'FUNCTION',11X,'DESIGN VECTOR'/)
79 GOTD 45
80 IF(KFEAS-2)80,80,78
81 WRITE(1,81)
FORMAT(/2X,'NO',4X,'STEP SIZE',7X,'UNAugMENTED',7X,'FUNCTION',
112X,'DESIGN VECTOR')
45 CONTINUE
KC1=1
KC2=1
NUM=1
H=AF(2)
2 AAL=(AF(1)*(AL(2)^2-AL(3)^2)+AF(2)*(AL(3)^2-AL(1)^2)+
1AF(3)*(AL(1)^2-AL(2)^2))/(2*(AF(1)*(AL(2)-AL(3))+
61AF(2)*(AL(3)-AL(1))+AF(3)*(AL(1)-AL(2))))
DO 6 J=1,NN
ZZ(J)=X(J)+AAL*S(J)
CALL FUNCT(ZZ,F)
CALL MODIFY(F)
76 IF(IDUT-1)76,76,77
82 IF(NC)82,82,83
52 WRITE(1,52)NUM,AAL,F,(ZZ(K),K=1,NN)
FORMAT(2X,I2,2(1PE14.6),4(7(1PE14.6),/,32X))
GOTD 77
83 IF(KFEAS-2)84,84,82
84 WRITE(1,85)NUM,AAL,UF,F,(ZZ(I),I=1,NN)
85 FORMAT(2X,I2,3(1PE14.6),8(6(1PE14.6)/46X))
RE-ESTIMATION OF THE RANGE LIMITS, SHUFFLING OF COORDINATES OF
THE THREE POINTS.
77 IF(ABS(F-H).LE.ACC)GOTO 7
8 IF(NUM-LIM)14,15,15
14 NUM=NUM+1
IF(AF(2)-F)9,55,10
9 H=AF(2)
IF(AAL-AL(2))17,7,18
17 AF(1)=F
AL(1)=AAL
IF(KC2-3)23,25,25
23 KC1=1
KC2=KC2+1
GOTD 2
18 AF(3)=F
AL(3)=AAL
IF(KC1-3)24,21,21
24 KC2=1
KC1=KC1+1
GOTD 2
10 H=F
11 IF(AAL-AL(2))11,7,12
AF(3)=AF(2)
AF(2)=F
AL(3)=AL(2)
AL(2)=AAL

```

```

12 AF(1)=AF(2)
   AF(2)=F
   AL(1)=AL(2)
   AL(2)=AAL
   IF(KC2-3)22,25,25
22 KC1=1
   KC2=KC2+1
   GOTD 2
C DETERMINATION OF NEW COORDINATES FOR THE FIRST POINT FOR
CC FOR FASTER CONVERGENCE
21 IF(IJUT.EQ.1)WRITE(1,92)
92 FORMAT(* CONVERGENCE SCHEME CALLED FROM 21 *)
   TLT=AL(2)-AL(1)
   TX=AL(3)-AL(2)
   KNN=1
   ST1=0.
   ST=KNN*TX
   ST1=ST1+ST
   IF(ST1-TLT)41,2,2
41 AAL=AL(2)-ST
   DO 27 I=1,NN
27 ZZ(I)=X(I)+AAL*S(I)
   CALL FUNCT(ZZ,F)
   CALL MODIFY(F)
   IF(F-AF(2))28,28,42
28 AF(2)=F
   AL(2)=AAL
   KNN=KNN+1
   GOTD 30
42 AF(1)=F
   AL(1)=AAL
   GOTD 2
C DETERMINATION OF NEW COORDINATES FOR THE THIRD POINT FOR
CC FASTER CONVERGENCE
25 IF(IJUT.EQ.1)WRITE(1,93)
93 FORMAT(* CONVERGENCE SCHEME CALLED FROM 25 *)
   TLT=AL(3)-AL(2)
   TX=AL(2)-AL(1)
   KNN=1
   ST1=0.
   ST=KNN*TX
   ST1=ST1+ST
   IF(ST1-TLT)39,2,2
39 AAL=AL(2)+ST
   DO 31 I=1,NN
31 ZZ(I)=X(I)+AAL*S(I)
   CALL FUNCT(ZZ,F)
   CALL MODIFY(F)
   IF(F-AF(2))32,32,40
32 AF(2)=F
   AL(2)=AAL
   KNN=KNN+1
   GOTD 34
40 AF(3)=F
   AL(3)=AAL
   GOTD 2
15 IF(IJUT.GT.2)GOTO 7
   WRITE(1,1) LIM,KAL
1 FORMAT(* ACCURACY NOT ACHIEVED IN ',I2,' ITERATIONS OF'
   ',I2,' th 1.0 CYCLE *)
   GOTD 88
7 IF(IJUT.GT.2)GOTO 88
   WRITE(1,89)NUM
89 FORMAT(' CONVERGENCE ACHIEVED IN ',I2,' ITERATIONS')
   GOTD 57
55 NMIN=NUM-1
   ALTER HERE
57 WRITE(1,90)AF(2),F
90 FORMAT(5X,'AF(2)',10X,'F'/2(1PE14.6))
88 IF(AF(2)-F)35,35,38
35 F=AF(2)
   AAL=AL(2)
   DO 37 I=1,NN
37 ZZ(I)=X(I)+AL(2)*S(I)
   IF(NC)38,38,150
150 CALL FUNCT(ZZ,F)
   CALL MODIFY(F)
38 DO 75 I=1,NN
75 X(I)=ZZ(I)
   IF(IJUT.GT.2)RETURN
   WRITE(1,95)AAL,AL(2),(ZZ(I),I=1,NN)

```

```

=====
SUBROUTINE GOLD(X,S,NN,XL,XH,F)
=====
DETERMINATION OF MINIMUM IN SEARCH DIRECTION USING
GOLDEN SECTION METHOD
X -- DESIGN VECTOR
NN -- ORDER OF DESIGN VECTOR
S -- SEARCH DIRECTION VECTOR
XL -- LOWER LIMIT FOR MINIMUM
XH -- HIGHER LIMIT FOR MINIMUM
LIM -- MAX NUMBER OF ITERATIONS PERMITTED
F -- FUNCTION VALUE AT THE MINIMUM
ACC -- ACCURACY DESIRED

```

```

COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT

```

```

COMMON/PENFUN/R,UF

```

```

COMMON/MIN/AAL

```

```

COMMON/NEW/FF,ACC,IGRAD

```

```

COMMON/ALARM/KAL,JAW

```

```

DIMENSION X(NN),S(50),AL(3),YY(3)

```

```

DIMENSION ZZ(25)

```

```

IF(IDUT-1)37,37,38

```

```

WRITE(1,39)

```

```

FORMAT(/,'GOLDEN SECTION RESULTS'/)

```

```

IF(NC)85,85,86

```

```

WRITE(1,40)

```

```

FORMAT(/2X,'STEP SIZE',9X,'FUNCTION',10X,'DESIGN VECTOR')

```

```

GO TO 38

```

```

IF(KFEAS-2)87,87,85

```

```

WRITE(1,88)

```

```

FORMAT(/2X,'STEP SIZE',6X,'UNAUGMENTED',4X,'FUNCTION',13X,
1'DESIGN VECTOR')

```

```

CONTINUE

```

```

KZ=1

```

```

B=XH-XL

```

```

AL(1)=XL+.618*B

```

```

AL(2)=XH-.618*B

```

```

DO 71 J=1,2

```

```

DO 70 K=1,NN

```

```

ZZ(K)=X(K)+ABS(AL(J))*S(K)

```

```

CALL FUNCT(ZZ,F)

```

```

CALL MODIFY(F)

```

```

YY(J)=F

```

```

IF(IDUT-1)80,80,71

```

```

IF(NC)89,89,90

```

```

WRITE(1,52) AL(J),YY(J),(ZZ(K),K=1,NN)

```

```

FORMAT(2D14.6,4(7D14.6,/,28X))

```

```

GO TO 71

```

```

IF(KFEAS-2)92,92,89

```

```

WRITE(1,91) AL(J),UF,YY(J),(ZZ(K),K=1,NN)

```

```

FORMAT(3D14.6,5(6D14.6,/,42X))

```

```

CONTINUE

```

```

IF(ABS(YY(2)-YY(1))-ACC)15,15,16

```

```

IF(KZ-LIM)25,26,26

```

```

KZ=KZ+1

```

```

DETERMINATION OF NEW LIMITS OF RANGE

```

```

IF(YY(2)-YY(1))18,15,17

```

```

XH=AL(1)

```

```

YY(1)=YY(2)

```

```

AL(1)=AL(2)

```

```

J=2

```

```

GO TO 31

```

```

XL=AL(2)

```

```

YY(2)=YY(1)

```

```

AL(2)=AL(1)

```

```

J=1

```

```

B=XH-XL

```

```

IF(J-1)32,32,33

```

```

AL(1)=XL+.618*B

```

```

DO 34 K=1,NN

```

```

ZZ(K)=X(K)+AL(1)*S(K)

```

```

GO TO 35

```

```

AL(2)=XH-.618*B

```

```

DO 36 K=1,NN

```

```

ZZ(K)=X(K)+AL(2)*S(K)

```

```

CALL FUNCT(ZZ,F)

```

```

CALL MODIFY(F)

```

```

YY(J)=F

```

```

IF(IDUT-1)84,84,85

```

```

IF(NC)93,93,94

```

```

WRITE(1,52) AL(J),YY(J),(ZZ(K),K=1,NN)

```

```

GO TO 45

```

```

IF(KFEAS-2)95,95,93

```

```

WRITE(1,93) AL(J),UF,YY(J),(ZZ(K),K=1,NN)

```

```

C      J=3
C      COMPUTATION OF THE MID-POINT TO CONFIRM MINIMUM AND
C      SATISFACTION OF ACCURACY CRITERION
72     DO 72 K=1,NN
        ZZ(K)=X(K)+ABS(AL(3))*S(K)
        CALL FUNCT(ZZ,F)
        CALL MODIFY(F)
        YY(J)=F
        IF(IOUT-1)82,82,83
82     IF(NCD)96,96,97
96     WRITE(1,52) AL(J),YY(J),(ZZ(K),K=1,NN)
        GO TO 83
97     IF(KFEAS-2)98,98,96
98     WRITE(1,91) AL(J),UF,YY(J),(ZZ(K),K=1,NN)
        IF(ABS(YY(3)-YY(1))-ACC)19,19,20
20     XL=AL(2)
        XH=AL(1)
        GO TO 61
19     AAL=ABS(AL(3))
        GO TO 27
26     IF(YY(2)-YY(1))76,76,77
76     YY(3)=YY(2)
        J=2
        GO TO 78
77     YY(3)=YY(1)
        J=1
78     DO 79 K=1,NN
79     ZZ(K)=X(K)+AL(J)*S(K)
        WRITE(1,1) LIM,KAL
1     FORMAT(* ACCURACY NOT ACHIEVED IN ',I3,' ITERATIONS
        1 OF ',I3,' to 1.D CYCLE *')
27     DO 75 J=1,NN
75     X(J)=ZZ(J)
        F=YY(3)
        RETURN
        END

```

```

=====
SUBROUTINE UNIV(MINIM,X,TT,NN,F)
=====
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
MINIM -- CHOICE OF 1.D SEARCH TECHNIQUE
MINIM=1 GOLDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
TT -- INITIAL STEP SIZE
ACC -- ACCURACY DESIRED
F -- OPTIMUM VALUE OF OBJECTIVE FUNCTION
.....
COMMON/INCDVC/G
COMMON/CONVEC/G
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,IDUT,NPERT
COMMON/NEW/FF,ACC,IGRAD
COMMON/ALARM/KAL,JAW
DIMENSION X(NN),AL(3),AF(3),Z(2)
DIMENSION S(50),G(25),GO(25)
KTK=1
TT1=TT
KAL=0
CALL FUNCT(X,F)
CALL MODIFY(F)
Z(1)=F
IF(NC)77,77,78
DO 79 J=1,NC
GO(J)=G(J)
CONTINUE
N=1
KAL=KAL+1
S(N)=1.
DO 11 J=1,NN
IF(J-N)12,11,12
S(J)=0.
CONTINUE
CALL RANGE(NN,NQUIT,NREP,TT,TT1,X,S,AF,AL,XH,XL,Z)
GOTO (21,30,31)NREP
GOTO (30,21,21)NQUIT
GOTO (900,901)MINIM
CALL GOLD(X,S,NN,XL,XH,F)
GOTO 903
CALL QUAD(X,S,NN,AL,AF,F)
IF(NC)80,80,81
DO 82 J=1,NC
GO(J)=G(J)
Z(2)=F
IF(KTK-3)83,84,84
IF(ABS(Z(2)-Z(1))-ACC)84,84,21
N=N+1
Z(1)=Z(2)
IF(N-NN)22,22,25
IF(NV-1)20,20,32
IF(NQUIT-3)27,20,20
CONVERGENCE RULE
IF(NPERT-1)20,20,85
CALL CONVRG(X,S,Z,NN,NC,KTK,HK,F)
GOTO (27,86,187,20)KTK
KAL=KAL+1
GOTO 87
PRINT OUTPUT
F=Z(2)
IF(IDUT.GT.4)GOTO 75
WRITE(1,3)
FORMAT('POWER'S UNIVARIATE METHOD USED ')
IF(IDUT.GT.5)GOTO 76
WRITE(1,177)KAL
FORMAT('NO OF 1.D CYCLES ',I3)
IF(IDUT.GT.3)GOTO 76
WRITE(1,211) (X(J),J=1,NN)
FORMAT(' [ X ] := ',9(1PE14.6))
WRITE(1,4) Z(2),ACC
FORMAT(' [ F ] := ',1PE14.6// 'ACCURACY ACHIEVED ',F9.5)
F=Z(2)
TT=TT1
RETURN
END

```

```

=====
SUBROUTINE STEEP(MINIM,X,TT,NN,F) !! STEEPEST DESCENT METHOD !!
=====
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
MINIM -- CHOICE OF 1.D SEARCH TECHNIQUE
MINIM=1 GOLDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
IT -- INITIAL STEP SIZE
ACC -- ACCURACY DESIRED
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE NCE
DIFFERENCE COMPUTATION OF THE GRADIENTS
IGRAD -- CHOICE OF GRADIENT TECHNIQUE
IGRAD=1 USE PACKAGE ROUTINE
IGRAD=2 USE USER ROUTINE
F -- OPTIMUM VALUE OF OBJECTIVE FUNCTION
.....
COMMON /CONVEC/G
COMMON /INCDVC/GD
COMMON /ALWAYS/TMAX,METHOD,KFEAS,LIM,VC,NI,IDUT,NPERT
COMMON /NEW/FF,ACC,IGRAD
COMMON /ALARM/KAL,JAW
DIMENSION X(NN),AL(3),AF(3),Z(2)
DIMENSION S(50),G(25),GD(25)
KTK=1
TT1=TT
KAL=0
45 CALL FUNC(X,F)
CALL MODIFY(F)
Z(1)=F
78 IF(NC) 77,77,78
79 DO 79 J=1,NC
77 GO(J)=G(J)
10 CONTINUE
KAL=KAL+1
700 GOTD (700,701)IGRAD
CALL GRAD1(X,S,SUM,NN,F)
701 GOTD 703
703 CALL GRADU(X,S,SUM)
S=SQR(SUM)
300 DO 300 J=1,NN
40 S(J)=S(J)/S
13 CALL RANGE(NN,NQUIT,NREP,TT,TT1,X,S,AF,AL,XH,XL,Z)
12 GOTD (11,12,13)NREP
900 GOTD (12,41,41)NQUIT
GOTD (900,901)MINIM
CALL GOLD(X,S,NN,XL,XH,F)
901 GOTD 903
903 CALL QUAD(X,S,NN,AL,AF,F)
81 IF(NC) 80,80,81
82 DO 82 J=1,NC
80 GO(J)=G(J)
11 Z(2)=F
CONTINUE
514 IF(KTK-3) 514,43,43
42 IF(ABS(Z(2)-Z(1))-ACC) 43,43,42
Z(1)=Z(2)
F=Z(2)
GOTD 10
43 CONVERGENCE RULE
44 IF(NPERT-1) 41,41,44
CALL CONVRG(X,S,Z,NN,NC,KTK,HK,F)
146 GOTD (10,45,146,41)KTK
KAL=KAL+1
GOTD 45
C PRINT OUTPUT
41 F=Z(2)
IF(IDUT.GT.4) GOTD 76
WRITE(1,203)
76 FORMAT('STEEPEST DESCENT METHOD USED ')
IF(IDUT.GT.5) GOTD 73
176 WRITE(1,176)KAL
FORMAT('NO OF 1.D CYCLES ',I4)
IF(IDUT.GT.3) GOTD 73
WRITE(1,4) (X(J),J=1,NN)
WRITE(1,204) Z(2),ACC
4 FORMAT(' [ X ] := ',9(1PE14.6))
204 FORMAT(' [ F ] := ',1PE14.6//, 'ACCURACY ACHIEVED ',F9.6)
73 TT=TT1
RETURN

```



```

F1=HK-2.*FMIN(NN)+F
F2=(HK-FMIN(NN)-FM)*2
F3=.5*FM*(HK-F)*2
ZZ=F1+F2-F3
IF(ZZ)34,32,32
SUM=J.
DO 35 J=1,NN
S(J)=X(J)-XO(J)
SUM=SUM+S(J)*2
SUM=SQR(SUM)
DO 36 J=1,NN
S(J)=S(J)/SUM
IF(NX-NN)40,41,41
NN1=NN-1
DO 37 J=1,NN1
IF(J-NX)37,38,38
DO 39 I=1,NN
SCD(J,I)=SCD(J+1,I)
CONTINUE
DO 42 J=1,NN
SCD(NN,J)=S(J)
I=NN
K=2
KIT=2
KAL=KAL+1
GOTO 26
SHUFFLE THE DIRECTION MATRIX TO GET SEARCH STARTED
NNI=NN-1
GOTO (59,72,72),KI
IF(KIT-NN)44,12,12
IF(KIT-NN1)60,60,52
IF(I-NN)51,52,52
HK=F
Z(1)=F
GOTO 33
DO 46 J=1,NN
SCD(NN+1,J)=SCD(I,J)
DO 47 J=1,NN
SCD(I,J)=SCD(I+1,J)
DO 48 J=1,NN1
DO 48 JK=1,NN
SCD(I+J,JK)=SCD(I+J+1,JK)
KIT=KIT+1
GOTO 24
CHECK FOR CONVERGENCE
IF(NPERI-1)12,12,73
CALL CONVRG(X,S,Z,NN,NC,KTK,HK,F)
GOTO(77,69,76,12)KTK
DO 75 J=1,NN
XO(J)=X(J)
KAL=KAL+1
GOTO 26
DO 78 J=1,NN
DO 78 I=1,NN
SCD(I,J)=0.
DO 79 I=1,NN
DO 79 J=1,NN
IF(I-J)79,80,79
SCD(I,J)=1.
CONTINUE
KI=1
KIT=1
KTK=1
I=2
K=1
DO 82 J=1,NN
SCD(1,J)=S(J)
GOTO 24
PRINT OUTPUT
F=Z(2)
IF(1OUT.GT.4)GOTO 173
WRITE(1,58)
FORMAT('POWELL'S METHOD [ CONDIR ] USED ')
IF(1OUT.GT.5)GOTO 55

```

```

=====
SUBROUTINE CONGRA(MINIM,X,TT,NN,F) !! FLETCHER-REEVES METHOD !!
=====
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
MINIM -- CHOICE OF 1.D SEARCH TECHNIQUE
MINIM=1 GOLDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
TT -- INITIAL STEP SIZE
ACC -- ACCURACY DESIRED
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE
DIFFERENCE COMPUTATION OF THE GRADIENTS
IGRAD -- CHOICE OF GRADIENT ROUTINE
IGRAD=1 USE PACKAGE ROUTINE
IGRAD=2 USE USER ROUTINE
F -- OPTIMUM VALUE OF OBJECTIVE FUNCTION

COMMON/CONVEC/G
COMMON/INCDVC/GO
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IOUT,NPERT
COMMON/NEW/FF,ACC,IGRAD
COMMON/ALARM/KAL,JAW
DIMENSION X(NN),AF(3),AL(3),Z(2)
DIMENSION A(25),S(50),SD(50),G(25),GD(25)
TT1=TT
KTK=1
KAL=0
CALL FUNC(X,F)
CALL MODIFY(F)
IF(NC) 77,77,78
DO 79 J=1,NC
GD(J)=G(J)
CONTINUE
HK=F
Z(1)=F
KAL=KAL+1
GOTO (700,701) IGRAD
CALL GRAD1(X,S,SUM,NN,F)
GOTO 703
CALL GRADU(X,S,SUM)
IST=0
B=SUM
SUM=SQRT(SUM)
DO 111 J=1,NN
SD(J)=S(J)
S(J)=S(J)/SUM
CALL RANGE(NN,NQUIT,NREP,TT,TT1,X,S,AF,AL,XH,XL,Z)
GOTO (51,50,52) NREP
GOTO (50,511,11) NQUIT
GOTO (900,901) MINIM
CALL GOLD(X,S,NN,XL,XH,F)
GOTO 903
CALL QUAD(X,S,NN,AL,AF,F)
IF(NC) 80,80,81
DO 82 J=1,NC
GD(J)=G(J)
CONTINUE
Z(2)=F
IF(KTK-3) 53,92,92
IF(ABS(Z(2)-Z(1))-ACC) 11,11,12
IF(ABS(Z(2)-HK)-ACC) 92,92,512
HK=Z(2)
Z(1)=Z(2)
F=Z(2)
GOTO 513
Z(1)=Z(2)
F=Z(2)
IF(IST-NN+1) 20,20,513
IST=IST+1
DETERMINATION OF CONJUGATE GRADIENT DIRECTION.
GOTO (705,706) IGRAD
CALL GRAD1(X,S,SUM,NN,F)
GOTO 704
CALL GRAD(X,S,SUM)
BETA=SUM/3
B=SUM
DO 46 J=1,NN
S(J)=S(J)+BETA*SD(J)
SD(J)=S(J)
CONTINUE
SA=0
DO 47 I=1,NN
SA=SA+S(I)*2
DO 49 I=1,NN
S(I)=S(I)/SQRT(SA)

```

```

C      KAL=KAL+1
92     GO TO 48
516    CONVERGENCE CRITERION
      IF(NPERT-1)511,511,516
      CALL CONVRG(X,S,Z,NN,NC,GTK,HK,F)
148    GO TO (513,103,148,511)GTK
      KAL=KAL+1
      GO TO 48
C      PRINT OUTPUT
511    F=Z(2)
      IF(IJUT.GT.4)GO TO 76
      WRITE(1,104)
104    FORMAT('FLETCHER - POWELL METHOD (CONGRA) USED')
76     IF(IJUT.GT.5)GO TO 73
      WRITE(1,176)KAL
176    FORMAT('NO OF 1.0 CYCLES ',I4)
      IF(IJUT.GT.3)GO TO 73
      WRITE(1,211)(X(I),I=1,NN)
211    FORMAT(' [ X ] := ',9(1PE14.8))
205    WRITE(1,205)Z(2),ACC
73     FORMAT(' [ F ] := ',1PE14.6/'ACCURACY ACHIEVED ',F9.5)
      IT=ITI
      RETURN
      END

```

```

=====
SUBROUTINE DFPM(MINIM,X,TT,NN,F) !! DAVIDON-FLETCHER-POWELL'S !!
=====
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
MINIM -- CHOICE OF 1.D SEARCH TECHNIQUE
MINIM=1 GOLDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
TT-----INITIAL STEP SIZE
ACC-----ACCURACY DESIRED
FF-----PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE
DIFFERENCE COMPUTATION OF THE GRADIENTS
IGRAD-----CHOICE OF GRADIENT SUBROUTINE
IGRAD=1 USE PACKAGE SUBROUTINE
IGRAD=2 USE USER SUBROUTINE
F-----OPTIMUM VALUE OF OBJECTIVE FUNCTION
.....
COMMON/CONVEC/G1
COMMON/INCDVC/G0
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,VC,NI,IOUT,NPERT
COMMON/MIN/AAL
COMMON/NEW/FF,ACC,IGRAD
COMMON/ALARM/KAL,JAW
COMMON/PEVPUN/R,F1
DIMENSION X(NN),AL(3),AF(3),Z(2)
DIMENSION S(50),H(25,25),FX(25),FY(25),G(25)
DIMENSION SQ(25)
DIMENSION G0(25),G1(25)
START SEARCH IN STEEPEST DESCENT DIRECTION
TT1=TT
KTK=1
KAL=0
CALL FUNCT(X,F)
CALL MODIFY(F)
Z(1)=F
IF(NC) 77,77,78
DO 79 J=1,NC
G0(J)=G1(J)
CONTINUE
HK=F
KAL=KAL+1
DO 10 J=1,NN
DO 10 I=1,NN
IF(I-J) 12,11,12
H(I,J)=1.
GO TO 10
H(I,J)=0.
CONTINUE
GO TO (700,701)IGRAD
CALL GRAD1(X,S,SUM,NN,F)
GO TO 703
CALL GRADU(X,S,SUM)
SUM=SQRT(SUM)
DO 13 J=1,NN
FX(J)=-S(J)
SQ(J)=S(J)
S(J)=S(J)/SUM
CALL RANGE(NN,NQUIT,NREP,TT,TT1,X,S,AF,AL,XH,XL,Z)
GO TO(514,16,74)NREP
GO TO(16,122,121)NQUIT
GO TO(900,901)MINIM
CALL GJLO(X,S,NN,XL,XH,F)
GO TO 903
CALL QUAD(X,S,NN,AL,AF,F)
IF(NC) 80,80,81
DO 82 J=1,NC
G0(J)=G1(J)
Z(2)=F
CHECK ACCURACY AND RESTART CRITERION.
IF(KTK-3) 515,92,92
IF(IOUT.LE.2)WRITE(1,95)(Z(1),I=1,2)
FORMAT('VALUES OF Z(1) and Z(2)'/2(1PE14.6))
IF(ABS(Z(2)-Z(1))-ACC) 121,121,22
IF(ABS(Z(2)-HK)-ACC) 92,92,123
F=Z(2)
HK=F
Z(1)=Z(2)
IF(IOUT.LE.2)WRITE(1,90)
FORMAT('/' * ITERATION IS RESTARTED FOR NEW S(J) *')
GO TO 124
F=Z(2)
GO TO (705,706)IGRAD
CALL GRAD1(X,S,SUM,NN,F)
GO TO 704
FOR 1 VARIABLE METRIC

```

```

06 CALL GRADU(X,S,SUM)
04 DO 14 J=1,NN
4 G(J)=-S(J)-FX(J)
SSUM=0.
DO 54 I=1,NN
DO 55 J=1,NN
SSUM=SSUM+H(I,J)*G(J)
FY(I)=SSUM
SSUM=0.
CONTINUE.
A2=0.
DO 52 J=1,NN
A2=A2+SQ(J)*G(J)
A2=AAL/A2
DO 50 I=1,NN
DO 50 J=1,NN
H(I,J)=H(I,J)+A2*SQ(I)*SQ(J)
A2=0.
DO 57 J=1,NN
A2=A2+FY(J)*G(J)
DO 56 I=1,NN
DO 56 J=1,NN
H(I,J)=H(I,J)-FY(I)*FY(J)/A2
DO 53 J=1,NN
FY(J)=-S(J)
SSUM=0.
DO 60 I=1,NN
DO 61 J=1,NN
SSUM=SSUM+H(I,J)*FY(J)
S(I)=-SSUM
SQ(I)=S(I)
SSUM=0.
CONTINUE.
DETERMINATION OF VARIABLE METRIC DIRECTION.
DO 52 J=1,NN
FX(J)=FY(J)
DO 53 J=1,NN
SSUM=SSUM+S(J)^2
SSUM=SQRT(SSUM)
IF(SSUM)67,66,67
SSUM=1.
DO 64 J=1,NN
S(J)=S(J)/SSUM
Z(1)=Z(2)
KAL=KAL+1
GOTO 65
CONVERGENCE CRITERION
IF(NPERT-1)122,122,516
CALL CONVRG(X,S,Z,NN,NC,KTK,HK,F)
GOTO (124,103,125,122)KTK
KAL=KAL+1
PRINT OUTPUT
F=Z(2)
IF(1OUT.GT.4)GOTO 75
WRITE(1,5)
FORMAT('DAVIDON - FLETCHER - POWELL METHOD USED')
IF(1OUT.GT.5)GOTO 76
WRITE(1,85)KAL
FORMAT('NO OF 1.0 CYCLES ',I3)
IF(1OUT.GT.3)GOTO 76
WRITE(1,3)(X(J),J=1,NN)
FORMAT(' [ X ] := ',9(1PE14.6))
WRITE(1,4)Z(2)
FORMAT(' [ F ] := ',1PE14.6)
F=Z(2)
IT=IT1
NO=NQUIT
RETURN
END

```

```

=====
SUBROUTINE GRAD1(X,S,SUM,NN,F)
=====
THIS ROUTINE DETERMINES THE DERIVATIVES USING THE FINITE
DIFFERENCE METHOD
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
F -- INITIAL VALUE OF THE OBJECTIVE FUNCTION
SUM -- SUM OF THE SQUARES OF THE DERIVATIVES
S -- SEARCH DIRECTION VECTOR
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR THE
COMPUTATION OF GRADIENTS
.....
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IOUT,NPERT
COMMON/NEW/FF,ACC,IGRAD
DIMENSION X(NN),S(50),SS(2)
SS(1)=F
SUM=0.
DO 10 J=1,NN
IF(X(J))60,61,60
61 Q=FF
62 GOTO 62
Q=FF*X(J)
X(J)=X(J)+Q
CALL FUNCT(X,F)
CALL MODIFY(F)
SS(2)=F
X(J)=X(J)-Q
S(J)=(SS(1)-SS(2))/Q
SUM=SUM+S(J)**2
IF(SUM)11,12,11
10 SUM=1.
12 CONTINUE
11 RETURN
END

```

```

SUBROUTINE GRAD(X,S,SUM,NN,F,NC,DG)
THIS ROUTINE DETERMINES THE DERIVATIVES USING THE FINITE
DIFFERENCE METHOD
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
F -- INITIAL VALUE OF THE OBJECTIVE FUNCTION
SUM -- SUM OF THE SQUARES OF THE DERIVATIVES
S -- SEARCH DIRECTION VECTOR
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR THE
COMPUTATION OF GRADIENTS
DG -- MATRIX OF CONSTRAINT GRADIENTS
G,GJ -- CONSTRAINT VECTORS
NC -- NUMBER OF CONSTRAINTS
.....
COMMON/CONVEC/G
COMMON/INCDVC/GJ
COMMON/NEW/FF,ACC,IGRAD
DIMENSION X(NN),S(50),SS(2),DG(NC,NN)
DIMENSION GJ(25),G(25)
SS(1)=F
SUM=0.
DO 10 J=1,NN
IF(X(J))60,61,60
61 Q=FF
62 GOTO 62
Q=FF*X(J)
X(J)=X(J)+Q
CALL FUNCT(X,F)
CALL MODIFY(F)
SS(2)=F
X(J)=X(J)-Q
S(J)=(SS(1)-SS(2))/Q
DO 20 K=1,NC
20 DG(K,J)=(G(K)-GJ(K))/Q
10 SUM=SUM+S(J)**2
IF(SUM)11,12,11
12 SUM=1.
11 CONTINUE
RETURN
END

```

```

C =====
C SUBROUTINE MODIFY(F)
C =====
COMMON/PENFUN/R,UF
COMMON/CONVEC/G
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIMM,NC,N1,LDUT,NPERT
COMMON/NEW/FF,ACC,IGRAD
DIMENSION G(25)
IF(NC.EQ.0)RETURN
GOTO(1,2,3)KFEAS
IF(N1)13,13,14
DO 5 J=1,N1
IF(G(J))5,5,6
F=F+R*G(J)*2
CONTINUE
NE=N2-N1
IF(NE)3,3,11
DO 9 J=1,NE
F=F+R*G(N1+J)*2
GOTO 3
IF(N1)15,15,16
DO 7 J=1,N1
IF(G(J))8,7,7
F=F-R/G(J)
CONTINUE
NE=NC-N1
IF(NE)3,3,12
DO 10 J=1,NE
F=F+2*(N1+J)*2/SQRT(R)
RETURN
END

```

```

=====
SUBROUTINE CONSTR(NPENAL)
=====
THIS ROUTINE CHECKS FOR VIOLATION OF ANY CONSTRAINTS
NPENAL=1 CONSTRAINTS SATISFIED
NPENAL=2 CONSTRAINTS VIOLATED
ACC -- ACCURACY DESIRED
NC -- NUMBER OF CONSTRAINTS
N -- INDEX INDICATING NUMBER OF CONSTRAINTS VIOLATED
VCONST -- VECTOR GIVING INFORMATION ON CONSTRAINTS VIOLATED
COMMON/ALWAYS/TMAX,METHOD,KPEAS,LIM,NC,NI,IDUT,NPERT
COMMON/CONVEC/G
COMMON/CONVID/N,NCONST
COMMON/NEW/FF,ACC,IGRAD
DIMENSION G(25),NCONST(25)
N=1
NPENAL=1
IF(KPEAS-1)16,16,17
DO 10 J=1,NI
IF(G(J)-.1E-10)11,12,12
11 IF(ABS(G(J))-ACC)13,13,10
13 NCONST(N)=J
GOTO 14
12 NCONST(N)=J
NPENAL=2
IF(KPEAS-2)16,15,18
N=N+1
10 CONTINUE
GOTO 15
15 NPENAL=2
16 RETURN
END

```



```

=====
SUBROUTINE CONVRG(X,S,Z,NN,NC,KTK,ZK,F)
=====
COMMON/CDNVEC/G
COMMON/INCDVC/GO
COMMON/NEW/FF,ACC,IGRAD
DIMENSION G(25),GO(25)
DIMENSION X(NN),S(50),Z(2),A(25)
GOTO (90,91,92)KTK
HK=Z(2)
F=Z(2)
Z(1)=Z(2)
KTK=2
DO 93 J=1,NN
A(J)=X(J)
DO 94 J=1,NN
X(J)=1.01*X(J)
CALL FUNCT(X,F)
CALL CONSTR(NPENAL)
IF(NPENAL-1)80,80,511
Z2HK=Z(2)-HK
IF(ABS(Z2HK)-ACC)511,511,512
IF(Z(2)-HK)99,99,100
DO 95 J=1,NN
S(J)=A(J)-X(J)
DO 83 J=1,NN
X(J)=A(J)
CALL FUNCT(X,F)
CALL MODIFY(F)
DO 107 J=1,NC
GO(J)=G(J)
CONTINUE
Z(1)=HK
GOTO 101
DO 102 J=1,NN
S(J)=X(J)-A(J)
HK=Z(2)
Z(1)=Z(2)
SUM=0.
DO 96 J=1,NN
SUM=SUM+S(J)*2
SUM=SQR(SUM)
DO 97 J=1,NN
S(J)=S(J)/SUM
KTK=3
GOTO 80
IF(ABS(Z(2)-HK)-ACC)511,511,98
KTK=1
F=Z(2)
Z(1)=Z(2)
HK=Z(2)
ZK=HK
IF(NC)80,80,84
DO 85 J=1,NC
GO(J)=G(J)
GOTO 80
KTK=4
RETURN
END
=====

```

```

=====
SUBROUTINE IPENAL(MINIM,X,TT,NN,FC,F)
=====
THIS ROUTINE USES THE INTERIOR PENALTY FUNCTION TECHNIQUE
TO FIND THE CONSTRAINED OPTIMUM
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
NC -- NUMBER OF CONSTRAINTS
TT -- INITIAL STEP SIZE
ACC -- ACCURACY DESIRED
MINIM -- CHOICE OF 1.D SEARCH TECHNIQUE
MINIM=1 GOLDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
METHOD -- CHOICE OF MULTIVARIATE SEARCH TECHNIQUE
METHOD=1 UNIVARIATE SEARCH
METHOD=2 STEEPEST DESCENT
METHOD=3 CONJUGATE DIRECTIONS
METHOD=4 CONJUGATE GRADIENT
METHOD=5 VARIABLE METRIC METHOD
KFEAS=1 UNCONSTRAINED OR EXTERIOR PENALTY FUNCTION PROBLEM
KFEAS=2 UNCONSTRAINED PROBLEM USING INTERIOR PENALTY FUNCTION
KFEAS=3 CONSTRAINED PROBLEM USING FEASIBLE DIRECTION METHOD
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE
DIFFERENCE COMPUTATION OF THE GRADIENTS
R -- PENALTY FACTOR
F1 -- UNAUGMENTED OBJECTIVE FUNCTION
VREP=3 TERMINATE. SEARCH DIRECTION DOESN'T IMPROVE OBJECTIVE
FUNCTION OR VIOLATES A CONSTRAINT
FC -- FACTOR FOR INCREASING OR DECREASING "R"
=====
COMMON/PENFUN/R,F1
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT
COMMON/NEW/FF,ACC,IGRAD
COMMON/ADARM/KAL,JAW
COMMON/CONVEC/G1
DIMENSION N(2,25),CC(25),DD(25)
DIMENSION V(2),RR(2),X(NN)
DIMENSION DG(25,25),G(25),G1(25)
JAW=1
IF(IDUT.GT.5)GOTO 10
WRITE(1,302)R,JAW
FORNAT(40(' '),R := '1PE9.2,' CYCLE NO ',I3/40(' '))
PERFORM UNCONSTRAINED MINIMIZATIONS FOR TWO VALUES OF " R "
NR=1
GOTO (450,451,452,453,454)METHOD
CALL UNIV(MINIM,X,TT,NN,F)
GOTO 455
CALL STEEP(MINIM,X,TT,NN,F)
GOTO 455
CALL CONDIR(MINIM,X,TT,NN,F)
GOTO 455
CALL CONGRA(MINIM,X,TT,NN,F)
GOTO 455
CALL OFPM(MINIM,X,TT,NN,F)
IF(IDUT.GT.3.OR.IDUT.GT.5)GOTO 110
WRITE(1,235)(X(I),I=1,NN)
WRITE(1,76)F
FORMAT(' [ X ] := ',9(1PE14.6))
FORMAT(' [ F ] := ',1PE16.8)
CALL FUNCT(X,F)
CALL MODIFY(F)
F3=F
IF(IDUT.GT.5)GOTO 35
WRITE(1,61)F1
RR(NR)=R
V(NR)=F1
DO 40 J=1,NN
N(NR,J)=X(J)
IF(NR-2)13,14,14
VR=NR+1
R=F3*R
JAW=JAW+1
IF(IDUT.GT.5)GOTO 15
WRITE(1,302)R,JAW
GOTO 15
BEGIN THE EXTRAPOLATION SCHEME
IF(ABS(V(2)-V(1))-ACC)116,116,117
F=V(2)
GOTO 36
VIAL=1
SC=SQRT(RR(1))
SD=SQRT(RR(2))
DEN=SC-SD
IF(IDUT.GT.4)GOTO 70
WRITE(1,462)

```

```

452 FORMAT('EXTRAPOLATION RESULTS')
70 AA=(V(2)*SC-V(1)*SD)/DEN
BB=(V(1)-V(2))/DEN
DO 15 J=1,NN
CC(J)=(W(2,J)*SC-W(1,J)*SD)/DEN
16 DD(J)=(W(1,J)-W(2,J))/DEN
IF(IOUT.GT.4)GOTO 71
82 WRITE(1,400)
400 FORMAT('EXTRAPOLATION SCHEME IS [ f ] := AA+BB*R*.5 /
1 [ f ] predicted optimum is AA ')
401 WRITE(1,401) AA
FORMAT('AA := ',1PE16.8)
402 WRITE(1,402)
FORMAT('EXTRAPOLATION SCHEME IS X(I)=CC(I)+DD(I)*R*.5//
1 [ X ] predicted optimum design vector is CC(1)')
403 WRITE(1,403)(CC(K),K=1,NN)
71 FORMAT(9(1PE16.8))
25 IF(NTAL-1)25,25,26
R=FC*R
JAN=JAN+1
AAD=AA
FMIN=AA+BB*SQR(R)
DO 17 J=1,NN
17 DD(J)=CC(J)+DD(J)*SQR(R)
CALL FUNCT(DD,F)
CALL MODIFY(F)
F2=F
CALL CONSTR(NPENAL)
IF(NPENAL-1)56,56,85
56 IF(F3-F2)57,57,53
53 DO 53 J=1,NN
56 X(J)=DD(J)
IF(IOUT.GT.4)GOTO 57
WRITE(1,85)R
86 FORMAT('EXTRAPOLATED VECTOR VALID WITH NEW R ',1PE9.2)
GOTO 57
85 CALL FUNCT(CC,F)
CALL MODIFY(F)
F4=F
CALL CONSTR(NPENAL)
IF(NPENAL-1)59,59,90
90 IF(IOUT.GT.4)GOTO 57
93 WRITE(1,93)
FORMAT('* PREDICTED VECTORS INFEASIBLE *')
GOTO 57
59 IF(F3-F4)88,88,91
91 DO 92 J=1,NN
92 X(J)=CC(J)
WRITE(1,87)
87 FORMAT('PREDICTED VECTOR CC VALID')
GOTO 57
88 WRITE(1,89)
89 FORMAT('EXTRAPOLATED and PREDICTED OPTIMUM VECTORS INCREASE
1 THE FUNCTION SO UNSUITABLE')
57 IF(IOUT.GT.5)GOTO 72
WRITE(1,405)
405 FORMAT('NEW STARTING POINT')
WRITE(1,403)(X(J),J=1,NN)
WRITE(1,302)R,JAN
C PERFORM UNCONSTRAINED MINIMIZATION FOR NEW VALUE OF "R"
72 GOTO (456,457,458,459,460)METHOD
456 CALL UNIV(MINIM,X,TT,NN,F)
GOTO 461
457 CALL STEEP(MINIM,X,TT,NN,F)
GOTO 461
458 CALL CONDIR(MINIM,X,TT,NN,F)
GOTO 461
459 CALL CONGRA(MINIM,X,TT,NN,F)
GOTO 461
460 CALL OFPM(MINIM,X,TT,NN,F)
461 IF(IOUT.LT.3.OR.IOUT.GT.5)GOTO 111
WRITE(1,235)(X(I),I=1,NN)
WRITE(1,76)F
111 CALL FUNCT(X,F)
CALL MODIFY(F)
F3=F
IF(IOUT.GT.5)GOTO 73
WRITE(1,61)F1
61 FORMAT(' [ f ] unaugmented := ',1PE16.8)
73 IF(ABS(V(2)-F1)-ACC)33,33,37
C CHECK ACCURACY CRITERION. IF NOT SATISFIED RE-COMPUTE CONSTANTS
37 IF(ABS(F1-FMIN)-ACC)18,18,19
18 VTAL=NTAL+1
19 RR(1)=RR(2)
RR(2)=R

```

```

V(1)=V(2)
V(1)=V(2)
V(2)=F1
DO 54 J=1,NN
54  N(1,J)=N(2,J)
    N(2,J)=X(J)
    GOTO 144
26  IF(ABS(F1-AAD)-ACC)33,33,31
31  NIAL=1
    GOTO 25
33  F=F1
35  IF(1.DUT.GT.6)GOTO 74
    WRITE(1,235)(X(I),I=1,NN)
    WRITE(1,52)F,JAW,ACC
52  FORMAT(' [ F ] := ',1PE16.8// 'ND OF CONSTRAINED CYCLES',13
1    / 'ACCURACY ACHIEVED := ',1PE8.2)
74  RETURN
    END

```

```

=====
SUBROUTINE EXPEN(MINIM,X,TT,NN,FC,F)
=====
THIS ROUTINE DETERMINES THE CONSTRAINED OPTIMUM USING THE
EXTERIOR PENALTY FUNCTION TECHNIQUE
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
MINIM -- CHOICE OF THE DESIGN VECTOR
MINIM=1 GOLDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
FF -- INITIAL STEP SIZE
NC -- NUMBER OF CONSTRAINTS
ACC -- ACCURACY DESIRED
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE
DIFFERENCE COMPUTATION OF GRADIENTS
IGRAD -- CHOICE OF GRADIENT ROUTINE
IGRAD=1 USE PACKAGE ROUTINE
IGRAD=2 USE USER ROUTINE
METHOD -- CHOICE OF MULTIVARIATE SEARCH TECHNIQUE
METHOD=1 UNIVARIATE SEARCH
METHOD=2 STEEPEST DESCENT
METHOD=3 CONJUGATE DIRECTIONS
METHOD=4 CONJUGATE GRADIENT
METHOD=5 VARIABLE METRIC METHOD
KFEAS=1 UNCONSTRAINED OR EXTERIOR PENALTY FUNCTION PROBLEM
KFEAS=2 UNCONSTRAINED PROBLEM USING INTERIOR PENALTY FUNCTION
KFEAS=3 CONSTRAINED PROBLEM USING FEASIBLE DIRECTION METHOD
R -- PENALTY FACTOR
F1 -- UNAUGMENTED OBJECTIVE FUNCTION
FC -- FACTOR FOR INCREASING OR DECREASING " R "
=====
* J=JAN/PEIRUN/R,F1 .....
* J=JAN/ALWAYS/TMAX,METHOD,KFEAS,LIM,NC,NI,IDUT,NPERT
* J=JAN/NEW/FF,ACC,IGRAD
* J=JAN/ALARM/KAL,JAN
DIMENSION X(NN),CC(2)
J=1;JAN=1
IF(IDUT.GT.5)GOTO 3
WRITE(1,10)R,JAN
PERFORM UNCONSTRAINED MINIMIZATIONS FOR TWO VALUES OF R
GOTO (450,451,452,453,454),METHOD
CALL UNIV(MINIM,X,TT,NN,F)
GOTO 455
CALL STEEP(MINIM,X,TT,NN,F)
GOTO 455
CALL CONDIR(MINIM,X,TT,NN,F)
GOTO 455
CALL CONGRA(MINIM,X,TT,NN,F)
GOTO 455
CALL DFP(MINIM,X,TT,NN,F)
IF(IDUT.GT.5)GOTO 110
WRITE(1,7)(X(I),I=1,NN)
WRITE(1,235)F
FORMAT(' [ X ] := ',9(1PE14.6))
FORMAT(' [ F ] := ',1PE14.6)
CALL FUNC(X,F)
CALL MODIFY(F)
CC(J)=F1
IF(IDUT.GT.5)GOTO 1
WRITE(1,61)CC(J)
FORMAT(' [ F ] unaugmented := ',1PE16.8)
IF(J-2)11,12,12
R=FC*R
JAN=JAN+1
J=2
IF(IDUT.GT.5)GOTO 3
WRITE(1,10)R,JAN
GOTO 3
CHECK ACCURACY CRITERION. IF NOT SATISFIED PERFORM UNCONSTRAINED
MINIMIZATION FOR NEW VALUE OF R
IF(ABS(CC(1)-CC(2))-ACC)4,4,5
CC(1)=CC(2)
R=FC*R;JAN=JAN+1
IF(IDUT.GT.5)GOTO 3
WRITE(1,10)R,JAN
FORMAT(40(' ')/'R := ',1PE9.2,' CYCLE NO ',12,/40(' '))
GOTO 3
F=CC(2)
IF(IDUT.GT.6)GOTO 2
WRITE(1,7)(X(J),J=1,NN)
WRITE(1,8)CC(2),JAN,ACC
FORMAT(' [ F ] := ',1PE16.8//'NO OF CONSTRAINED CYCLES ',13/
1 // 'ACCURACY ACHIEVED ',1PE8.2)
2 RETURN
END

```

```

=====
SUBROUTINE FEAS(MINIM,X,TT,NN,W,NC,DG)
=====
THIS ROUTINE DETERMINES THE CONSTRAINED OPTIMUM USING
THE METHOD OF FEASIBLE DIRECTIONS
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
MINIM -- CHOICE OF 1.D SEARCH TECHNIQUE
MINIM=1 GGDEN SECTION
MINIM=2 QUADRATIC INTERPOLATION
IGRAD -- CHOICE OF GRADIENT ROUTINE
IGRAD=1 USE PACKAGE ROUTINE
IGRAD=2 USE USER ROUTINE
TT -- INITIAL STEP SIZE
ACC -- ACCURACY DESIRED
PP -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE
DIFFERENCE COMPUTATION OF GRADIENTS
N -- VECTOR OF PUSH - OFF FACTORS
NC -- NUMBER OF CONSTRAINTS
N -- INDEX INDICATING NUMBER OF CONSTRAINTS VIOLATED
NCONST -- VECTOR GIVING INFO ON CONSTRAINTS VIOLATED
DG -- MATRIX OF CONSTRAINT GRADIENTS
G,GD -- CONSTRAINT VECTORS
.....
COMMON/CONST/N,NCONST
COMMON/CONST/VEC/G
COMMON/CONST/INC/DVC/GD
COMMON/CONST/NEW/PP,ACC,IGRAD
DIMENSION X(NN),Z(2),AL(3),AF(3),W(NC),DG(NC,NN)
DIMENSION S(50),G(25),GD(25),NCONST(25)
DIMENSION A(25,50),B(25),C(50)
KJ=1
CALL FUNC(X,F)
Z(1)=F
DO 70 J=1,NC
GD(J)=G(J)
70 IT1=TT
GOTO (700,701)IGRAD
700 CALL GRAD(X,S,SUM,NN,F,NC,DG)
GOTO 703
701 CALL GRADU(X,S,SUM)
703 SUM=SQRT(SUM)
DO 111 J=1,NN
S(J)=S(J)/SUM
111 CALL RANGE(NN,NQUIT,NREP,TT,TT1,X,S,AF,AL,XH,XL,Z)
24 GOTO (10,11,12)NREP
11 GOTO (900,901)MINIM
900 CALL GJLO(X,S,NN,XL,XH,F)
GOTO 903
901 CALL QUAD(X,S,NN,AL,AF,F)
903 DO 40 J=1,NC
40 GD(J)=G(J)
Z(2)=F
IF(ABS(Z(2)-Z(1))-ACC)122,122,22
Z(1)=Z(2)
F=Z(2)
GOTO 23
12 F=Z(2)
GOTO (122,13,122)NQUIT
13 IF(KJ-2)80,81,81
80 F1=Z(2)
KJ=2
GOTO 82
81 F2=Z(2)
IF(ABS(F2-F1)-ACC)122,122,83
83 F1=F2
82 IX=N-1
CONSTRAINTS IN EFFECT. COMPUTE USABLE FEASIBLE DIRECTION
USING LINEAR PROGRAMMING
N1=N+NN
N2=NN+1+N1
N3=NN+1
DO 79 I=1,N1
DO 79 J=1,N2
79 A(I,J)=0.
DO 31 I=1,NN
DO 31 J=1,NN
IF(I-J)31,33,31
33 A(IX+1+I,J)=1.
31 CONTINUE
DO 35 J=1,NN
35 B(IX+1+J)=2.
GOTO (705,706),IGRAD
705 CALL GRAD(X,S,SUM,NN,F,NC,DG)
GOTO 704

```

```

706 CALL GRADJ(X,S,SUM)
704 DO 50 J=1,IX
K=VCONST(J)
DO 51 I=1,NN
51 A(J,I)=DG(K,I)
SUM=0.
DO 52 I=1,NN
52 SUM=SUM+DG(K,I)
B(J)=SUM
CONTINUE
DO 55 J=1,IX
K=VCONST(J)
56 A(J,N3)=W(K)
CONTINUE
IX=IX+1
A(IX,N3)=1.
DO 54 I=1,N1
DO 54 J=1,N1
IF(I-J)54,57,54
57 A(I,N3+J)=1.
54 CONTINUE
DO 74 J=1,N2
74 C(J)=0.
C(N3)=-1.
SUM=0.
DO 30 J=1,NN
30 A(IX,J)=-S(J)
SUM=SUM-S(J)
B(IX)=SUM
CALL LINEAR(A,B,C,S,N1,N2,N3)
DO 73 J=1,NN
78 S(J)=S(J)-1.
SUM=0.
DO 59 J=1,NN
59 SUM=SUM+S(J)**2
SUM=SQR(SUM)
IF(SUM)61,62,61
62 SUM=1.
61 DO 60 J=1,NN
60 S(J)=S(J)/SUM
GOTO 24
122 WRITE(1,5)(X(J),J=1,NN)
5 FORMAT(' [ X ] := ',9(1PE14.6))
4 WRITE(1,4)Z(2)
73 FORMAT(' [ F ] := ',1PE14.6)
RETURN
END

```



```

=====
SUBROUTINE LINEAR(A,B,C,Z,N1,N2,N3)
=====
SOLUTION OF A L.P. PROBLEM BY SIMPLEX METHOD
A -- COEFFICIENT MATRIX
B -- VECTOR OF QUANTITIES ON THE RIGHT SIDE OF THE EQUATIONS
C -- SENSITIVITY COEFFICIENTS
N1 -- NUMBER OF INEQUALITIES
N2 -- TOTAL NUMBER OF VARIABLES := REAL + SLACK
N3 -- NUMBER OF REAL VARIABLES
COMMON/ALWAYS/TMAX,METHOD,KFEAS,LIM,VC,NI,IJUT,NPERT
DIMENSION A(25,50),B(25),C(50),E(25),CB(50),Z(50),NBASIS(25)
IF(IJUT-1)101,101,102
WRITE(1,95)
101
95
FORMAT(' LINEAR PROGRAMMING RESULTS')
WRITE(1,66)N1,N2,N3
66
FORMAT(' INEQUALITIES =',I4,' NOS. OF VARIABLES =',I4,
1,' REAL VARIABLES =',I4)
C
102
FORM INITIAL BASIS
DO 10 J=1,N1
NBASIS(J)=N3+J
10
CB(J)=C(N3+J)
IF(IJUT-1)97,97,87
97
WRITE(1,60)
60
FORMAT(' NBASIS',5X,' CB',8X,' B',60X,' V')
DO 94 J=1,N1
94
WRITE(1,61)NBASIS(J),CB(J),B(J),(A(J,K),K=1,N2)
CONTINUE
CHECK IF ALL B,S ARE POSITIVE. IF NOT RESHUFFLE BASIS.
87
DO 74 J=1,N1
IF(B(J))73,74,74
73
IF(A3S(B(J))-1.E-10)89,89,90
89
B(J)=0.
GO TO 74
90
DO 72 K=1,N2
72
A(J,K)=-A(J,K)
B(J)=-B(J)
M=1
DO 75 K=1,N2
IF(A(J,K))91,91,77
IF(A(J,K)-1.E-10)91,91,92
91
M=M+1
76
CONTINUE
92
IF(M-N2)93,93,47
93
D=A(J,K)
MC=K
DO 78 K=1,N2
IF(A(J,K))78,78,79
IF(D-A(J,K))80,78,78
80
D=A(J,K)
MC=K
78
CONTINUE
NBASIS(J)=MC
CB(J)=C(MC)
DO 85 K=1,N2
85
A(J,K)=A(J,K)/D
B(J)=B(J)/D
DO 81 K=1,N1
IF(K-J)82,81,82
82
D=A(K,MC)
DO 83 I=1,N2
83
A(K,I)=A(K,I)-D*A(J,I)
B(K)=B(K)-D*B(J)
CONTINUE
74
CONTINUE
DO 85 J=1,N1
86
IF(B(J))87,86,85
CONTINUE
SUM=0.
DO 56 J=1,N1
C
56
SELECT COLUMN TO BE INCLUDED IN THE BASIS
SUM=SUM+B(J)*CB(J)
F=SUM
DO 12 K=1,N2
SUM=0.
DO 11 J=1,N1
11
SUM=SUM+A(J,K)*CB(J)
12
Z(K)=SUM-Z(K)
CONTINUE
KM=1
24
N=1
DO 40 K=1,N2
44
IF(Z(K)-1.E-10)44,44,41
N=N+1

```



```

40 CONTINUE
41 IF(N=N2)42,42,43
42 D=Z(K)
MC=K
DO 13 J=1,N2
IF(Z(J))13,13,30
30 IF(D-Z(J))14,13,13
14 D=Z(J)
MC=J
13 CONTINUE
C SELECT ROW TO BE REMOVED FROM THE BASIS.
DO 15 J=1,N1
IF(A(J,MC))71,71,95
95 IF(A(J,MC)-1.E-10)71,71,70
70 E(J)=B(J)/A(J,MC)
GOTO 15
71 E(J)=-1.
15 CONTINUE
M=1
DO 27 J=1,N1
IF(E(J))45,28,28
45 M=M+1
27 CONTINUE
28 IF(M=N1)46,46,47
46 D=E(J)
MR=J
DO 16 J=1,N1
IF(E(J))16,31,31
31 IF(D-E(J))16,16,17
17 D=E(J)
MR=J
16 CONTINUE
IF(K4-1)5,5,6
5 IF(LOUT-1)99,98,6
98 WRITE(1,60)
DO 54 J=1,N1
WRITE(1,61)NBASIS(J),CB(J),B(J),(A(J,K),K=1,N2)
61 FORMAT(2X,I4,3(10F10.4,/,26X))
64 CONTINUE
WRITE(1,63)F,(Z(K),K=1,N2)
63 FORMAT(2X,'F=' ,F10.4,/,2X,'Z(K)=' ,3(12F10.4,/,7X))
C FORMATION OF THE NEW SIMPLEX TABLE BY REPLACING ROW IN THE BASIS
C BY THE COLUMN. REPEAT THE CYCLE OF OPERATIONS.
6 CB(MR)=C(MC)
NBASIS(MR)=MC
D=Z(MC)
DO 23 K=1,N2
23 Z(K)=Z(K)-A(MR,K)*Q/A(MR,MC)
D=A(MR,MC)
DO 18 J=1,N2
18 A(MR,J)=A(MR,J)/D
B(MR)=B(MR)/D
DO 19 J=1,N1
IF(J-MR)21,19,21
21 D=A(J,MC)
DO 20 K=1,N2
20 A(J,K)=A(J,K)-D*A(MR,K)
B(J)=B(J)-D*B(MR)
19 CONTINUE
SUM=0.
DO 55 J=1,N1
55 SUM=SUM+B(J)*CB(J)
F=SUM
IF(LOUT-1)99,99,100
99 WRITE(1,60)
DO 75 J=1,N1
75 WRITE(1,61)NBASIS(J),CB(J),B(J),(A(J,K),K=1,N2)
CONTINUE
WRITE(1,63)F,(Z(K),K=1,N2)
100 KM=2
GOTO 24
C PRINT OUTPUT
43 SUM=0.
DO 22 J=1,N1
22 SUM=SUM+B(J)*CB(J)
F=SUM
DO 50 K=1,N2
50 Z(K)=0.
DO 51 J=1,N1
51 K=NBASIS(J)
Z(K)=B(J)
CONTINUE
IF(LOUT-4)103,103,52
103 WRITE(1,1)(Z(J),J=1,N3)
1 FORMAT('I X J :=' ,9(1PE14.6))

```

```
3      WRITE(1,3)F
      FORMAT(' [ F ] := ',1PE14.6)
      GO TO 52
47     WRITE(1,4)
4      FORMAT(' NO SOLUTION POSSIBLE')
52     CONTINUE.
      RETURN
      END
```

```

=====
SUBROUTINE START(X,NN,NC,TT,DG,IOUT)
=====
DETERMINES A FEASIBLE STARTING POINT
X -- DESIGN VECTOR
NN -- ORDER OF THE DESIGN VECTOR
IGRAD -- CHOICE OF GRADIENT ROUTINES
IGRAD=1 USE PACKAGE ROUTINE
IGRAD=2 USE USER ROUTINE
TT -- INITIAL STEP SIZE
FF -- PERCENTAGE CHANGE IN VARIABLE DESIRED FOR FINITE
DIFFERENCE COMPUTATION OF THE GRADIENTS
NC -- NUMBER OF CONSTRAINTS
G -- CONSTRAINT VECTOR
DG -- MATRIX OF CONSTRAINT GRADIENTS
=====
COMMON/CONVEC/G
COMMON/INCDVC/GO
COMMON/NEW/FF,ACC,IGRAD
DIMENSION X(NN),DG(NC,NN)
DIMENSION S(50),G(25),YX(25),GO(25)
DOUBLE PRECISION G,GO,D,DG,SUM,S,X,YX,AAL,AALO
COMPUTE CONSTRAINTS. DETERMINE CONSTRAINT MOST VIOLATED
CALL FUNCT(X,F)
IF(LOUT.GE.2)GOTO 13
WRITE(1,19)
WRITE(1,17)(X(J),J=1,NN)
WRITE(1,18)
WRITE(1,17)(G(J),J=1,NC)
V=1
DO 24 J=1,NC
GO(J)=G(J)
AAL=TT
DO 3 K=1,NC
IF(G(K))1,2,2
V=N+1
CONTINUE
IF(V-NC)7,7,10
D=G(K)
MC=K
DO 3 K=1,NC
IF(G(K)-D)3,3,6
D=G(K)
MC=K
CONTINUE
DETERMINE SEARCH DIRECTION. NORMAL TO CONSTRAINT MOST VIOLATED
GOTO (700,701)IGRAD
CALL GRAD(X,S,SUM,NN,F,NC,DG)
GOTO 703
CALL GRADU(X,S,SUM)
DO 11 J=1,NN
S(J)=-DG(MC,J)
SUM=0.
DO 20 J=1,NN
SUM=SUM+S(J)*2
SUM=SQRT(SUM)
IF(SUM)21,22,21
SUM=1.
CONTINUE
DO 23 J=1,NN
S(J)=S(J)/SUM
IF(LOUT.GE.2)GOTO 15
WRITE(1,35)
FORMAT('VALUES OF S')
WRITE(1,17)(S(J),J=1,NN)
DO 12 J=1,NN
YX(J)=X(J)+AAL*S(J)
CALL FUNCT(YX,F)
IF(LOUT.GE.2)GOTO 93
WRITE(1,19)
WRITE(1,17)(YX(J),J=1,NN)
WRITE(1,18)
WRITE(1,17)(G(J),J=1,NC)
WRITE(1,79)
FORMAT('VALUE OF G(MC) and GO(MC)')
WRITE(1,17)G(MC),GO(MC)
IF(G(MC)-GO(MC))27,27,30
DO 31 J=1,NN
YX(J)=X(J)+AALO*S(J)
DO 9 J=1,NN
X(J)=YX(J)
GOTO 28
DO 29 J=1,NC
GO(J)=G(J)
AALO=AAL

```

```

C      CHECK IF ALL CONSTRAINTS ARE SATISFIED. IF NOT REPEAT THE CYCLE)
32     IF(G(MC))32,14,14
33     DO 33 J=1,NN
34     X(J)=YX(J)
14     GO TO 13
15     AAL=2.*AAL
16     GO TO 15
10     IF(1DUT.GE.2)GO TO 95
94     WRITE(1,16)
16     FORMAT('FEASIBLE STARTING POINT')
17     WRITE(1,19)
19     FORMAT('VALUES OF X ARE')
17     WRITE(1,17)(X(J),J=1,NN)
18     FORMAT(9(1PE14.6))
18     WRITE(1,18)
18     FORMAT('VALUES OF G')
18     WRITE(1,17)(G(J),J=1,NC)
95     RETURN
END

```

A 83399

ME-1984-M-KUM-DEV.